DRAGON USER

The independent Dragon magazine

August 1987

Contents

Letters 2
DeltaDOS ... thoughts on Basic 42 ... autorun solution ... and plenty more.

News 4
Dragon show in September ... 67 Dragons in letterbox ... new light pen software ... modem for Dragon 64.

Dragonsoft 5
Prism modem package for the Dragon 64, by Hayton Software.

The Answer 6
Not only do we have the answer to the May competition, but we have the answer to the April competition.

Puzzle FORTH 8
J. B. Slinger investigates FORTH by untangling past Gordon Lee puzzles.

Winners and losers 15
Gordon Lee looks at some of the solutions to the March competition, and compares run times — from 10.96 seconds to 3 hours.

The Show goes on 16
Ken G. Smith finally makes it back from the 6809 Show and reports to HQ.

Down in the dumps 17
We asked for dumps, you sent us dumps. These three are for the Epson FX-85, OKI-80 Microline and Tandy CGP-155 plotter.

Dragon Answers 20
Sound to the cassette port ... connecting an Anadex printer ... Flex or OS-9?.

Super SAM 21
Treat your Synchronous Address Multiplexer gently — it doesn't mind being interrupted, but it hates being poked, says Matthew Lodge.

Electronic author 23
Philip Beed writes up Quickbeam's *Electronic Author.*

Two special offers 25
Three popular books of Dragon programs, no longer available from their original publishers, and a FREE GAME with the first 450 orders. ALSO *Dragon Home Accounts* for the Dragon 32, at less than half the original price.

Write: ADVENTURE 27
Peter Gerrard gets back to his parser and digresses into a little marketing advice.

Adventure trail 29
Retailing Basic games (why they don't), pokes for *Keys of the Wizard*.

Competition 31
Gordon Lee is looking for a mysterious factor, while Peaksoft are shelling out T-shirts.

Editorial

WALES, home of the Dragon. The very place for a summer holiday — as long as you stay indoors. No, seriously, it's not like that. It only rains in the mornings. Anyway, I'm off next week, so if you see a puff of smoke and flame along Hay Bluff, it won't be me, 'cos I'm not telling anyone where I am...

...just like The Expert, in fact. We would like to think that he's lost in the post, the postal strike started after he should have been here...

The best news this month is that Pulsar Software are planning a show in Rochdale later in the year. Somebody else has an eye on a London venue, and some of the NDUGs are putting on an informal show on the south coast. The future of the Dragon lies in bringing together suppliers, users and software writers in a setting which is substantial enough to draw people from all over Britain, and informal enough to accommodate moderate numbers in comfort. No more echoing halls.

And don't miss our special book offer. Dragon books are getting harder to find as big publishers look elsewhere. Now is a good time to buy out of print books at remainder prices. Have a look and see.

Telephone number
(All departments)
437-4343

Editor
HELEN ARMSTRONG

Production Editor
BARBORA HÁJEK

Editorial Secretary
ANNE MARIE ALLEN

Advertisement Manager
ATHENA PEERMAN

Administration
GABRIELLA PISCINA

Marketing Manager
HELEN PERRY

Managing Editor
PETER WORLOCK

Publishing Director
JENNY IRELAND

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues
ISSN 0265-177. Telex: 296275
Dragon User, 12/13 Little Newport Street, London WC2H 7PP
US address: c/o Business Press International, 205 East 42nd St, New York, NY 10017
Published by Scot Press Ltd.

Typesetting and Production by Artext Limited, London NW1.
Printed by Headley Brothers Ltd. Ashford, Kent
Registered at the Post Office as a newspaper.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

Basic extension

HAVING just read Roy Coates's review of *Basic 42* again, I decided to add some comments.

Firstly, Roy's review and the ads might lead some to suppose, on booting the system, that you could expect more memory space available to Basic than when operating in 32K mode. This is not the case, in fact, it is exactly the same. There are 3.75K extra available for machine code or data storage, but this is the area used by the utilities and it cannot be used if you want to use the non-destructive window 9.

Another point is that was with Compusense's *Hi Res*, once a character is drawn on the screen it is forgotten, so a high resolution screen dump is required if you want hard copy of your information. This is not a problem if you have a machine code dump routine, as the start and finish addresses of the current window are readily available and can be sent to the dump routine so that just the current window is printed. The keyboard response seems to be slower and I find that I lose characters. This seems to be alleviated by shortening the delay before autorepeat.

Because the system recognises PRINT@ and prints it relative to the window being used, running existing programs on *Basic 42* can give some unexpected screen displays. The best way round this is to use the FRAME command and define a window 32 characters wide and 16 deep, use this window and the display is restored. The use of windows gives a much more professional look to your programs, making it possible to display a program title at the top of the screen while all the action takes place below. If a command window is then used at the bottom of the screen for requesting inputs, keypresses etc., the result can be very impressive.

Finally, accepting commands in lower case letters may be very desirable when programming, but it can be infuriating when all INPUTs and LINE INPUTs are treated in the same way, ie converted to upper case. Unfortunately, this is exactly what happens. Help is at hand here, for when I contacted Bob Harris he advised that POKE 57685, 41 switches the facility off and POKE 57658, 15 puts it back on again. Once these were incorporated at the beginning and end of the program, the problem is solved.

After reading this you might think that I am somewhat disappointed with the package, but you would be wrong. Using the system is simplicity itself, a quick browse through the excellent manual and I was ready to go, and the overall effect was that it made my work look good and in doing so made me look good, and who could ask for more than that? I shall have to get the spool utility though, as at present I can make the tea while the machine dumps the hi-res screen.

Ken Smith
33 Glack Road
Deal
Kent
CT14 9ND

Every month we will be shelling out a game or two, courtesy of Microdeal, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What d'you think we are, mind readers?!

EXTRA PUFF

Delta users

DUDE stands for Delta Users Data Exchange. I initiated it after spending a long time struggling without information, professional software etc. As time passes it becomes increasingly obvious that the Dragon community is forming into specific user groups and it is natural that Delta users should also have a common base. The group is still young and has no definite structure, as I am still waiting to see if there is a good response from the details given in DU.

If there is, then it's likely that a NEWSDISK will be put together, as the OS-9 Group have done. This keeps costs down for the members and allows easy transfer of software. Having already contacted a few people, the initial response seems favourable and I have made several new 'friends' (if only over the phone!). I didn't realise just how helpful some people can be, not to mention the enthusiasm they have for Delta (it IS the best DOS!). The group will only thrive if other Delta users get in contact and show some interest, so write to me or phone on (0703) 38042 and find out what you're missing! If you don't need something now, you may do in the future, and you could make a valuable contribution.

Dave Martin, 44 Firgrove Road, Freemantle, Southampton, Hants SO1 3EN.

PS To all the people who 'registered' at the show, but have not contacted me since... you should be ashamed, get your writing pad out NOW!

THE user groups are the backbone of specialist Dragonisation just as *Dragon User* is the backbone of general Dragonisation. They form the networks which keep users in touch. I strongly suggest that Delta users get in touch with David to find out what is going on, even if they don't see how a group would be of immediate benefit. Then, if nothing else, someone would have a register where Delta people could get in touch.

Dragon chunnel

WITH regards to shows, has anyone thought of using Dover?

I know we have many Dragon owners in Belgium and Holland, not to mention France. If we now add those from the south of England we could have an international Dragon Show. I suppose it is all just a dream but it might make people take notice of us.

Ken G. Smith
33 Glack Road
Deal
Kent
CT14 9ND

Curve warped

THE euphoria of finding my name mentioned three times on Gordon Lee's *Winners and Losers* page June 1987 soon evaporated when two mistakes were spotted in my Dragon Curve program listing. If anyone has tried the program then it would have crashed almost instantly. The following corrections will rectify matters:

Line 60 had a bracket missing and should have read:

```
60 D = D + (S(C) = S(C + 1))
-(S(C)<>S(C + 1))
```

Line 150 included the number 570, the only 570 I know is house number, perhaps that was a hint, it should have read:

```
150 IF S(0) = 0 THEN D = 0:
GOTO 50
```

D.J.Gray
570 Acklam Road
Middlesbrough
Cleveland
TS5 8BE

AFTER everything else, too. Never mind. Let's hope this correction puts the curve straight, so to speak.

Revoxed

RECENTLY my Dragon 32 lost its voice. Occasionally it does return for brief periods, but is unpredictable. Since all the software still worked OK on it I was loath to send it to be repaired in case it 'died' on the operating table.

I had resigned myself to its silence until last week. I use a Saisho DR10 data recorder

which has a switchable monitor and LED indicator. I had just loaded a game, and forgot to press the STOP button, and started playing. I was surprised to see that the LED indicator was jumping in time with the expected sound effects. I switched on the monitor and was surprised to find the Dragon was using the cassette to speak to me. I have since played around and found the sequence to get the new voice is a) having loaded press STOP to release the head, b) put a blank tape in to keep the sensor pressed and press RECORD only, c) switch the monitor to on, and reduce the volume considerably. The Dragon sends all sound output to the data recorder.

I managed to find someone else with a fully working Dragon, who tried this and found it worked on his also. In fact the sound came both from the television and the cassette recorder, so you can have a stereo Dragon.... I believe the original problem with my Dragon is simply a dry joint somewhere, but now I don't have to worry about it.

Thank you for your excellent magazine. Long Live the Dragon.

M. Hall
103 Mandarin Way
Wyman's Brook
Cheltenham
Glos.
GL50 4RS

PS I persuaded my friend with the working Dragon to subscribe to *Dragon User*, do I get commission??

WELL er not as such, but if your friend finds he doesn't like us you can come and hide in the filing cabinet here for a bit. Um, can you type, by any chance?

Listings lamblasted

I have delayed renewing my subscription as I have become rather disappointed with the magazine recently as it appears to contain little but long pages of listings. The June editorial indicates some hope, so I will now renew.

B.P. Collins

STREWTH! It's a tough life trying to run a computer magazine without putting the odd six-page listing in, y'know.

OK, lets get back to work: the great printer issue is over and done, although we still have a stock of screen dumps which we shall be running in clumps. NOW we would like to hear from anyone who would like to write a review, description or educational piece on disc drives or disc operating systems. The hardware doesn't have to be new, but it must either be currently available or widely owned. Drop me a line and I will write back with details.

Autorun

LOOKING back in the life of the Dragon, people have always made hard work of autorunning a Basic program. Here is my short solution.

1) Load basic program as normal.
2) To save type:

```
PCLEAR 1 : CLS:?@235,
"LOADING!":
CSAVEM "Name", 166, 256 *
PEEK (27) + 255, 0:
POKE 25, 12: POKE 27, PEEK
(126): POKE 29, PEEK (27):
POKE 31, PEEK (27): RUN
```

This works by saving the direct command along with the Basic, so when reloaded it carries on with the command from where it left off causing the locations to be reset and a RUN command.

Paul Burgin
18 Moorfield Road
Fulwood
Sheffield
S10 4GS

Malta message

THROUGH the medium of your magazine Dragon owners have remained united, and help each other. It's with this in mind that I'm writing this letter. Others helped me, so I should help them.

In the book *Inside the Dragon* on page 207, I found a very important poke. How to print on the printer exactly what in the screen, regarding printer width, but it has a small mistake.

In short, the following Basic Direct (without line number) statements set up the printer width to be the same as the Dragon's display width, of 32 characters per line.

```
Poke &H148,0
Poke &H9B,32
LLIST
```

That has a mistake in it in the book, with a $ instead of the correct &.

I would like to take this opportunity to thank Pam D'Arcy who in the *Dragon User* of Sep. 86 pointed out another small mistake on page 215 of *Inside the Dragon* (ie, the cassette gap byte being $01 for 'continuous stream' whereas it is actually $00).

In my opinion *Inside the Dragon* is the best book for serious programmers, but to clear out the bugs and get an OK message, one still needs to take a good look at the *Dragon User*. My compliments to the Editor and all the staff.

Joe Brincat
73 Annunciation Street
Hamrun
Malta

JOE is another of those Dragoners who is a really useful human being, and writes us nice letters, too.

Main man

I have read your magazine since it first came out, and I am amazed at the amount of detrimental rubbish stated about both the Dragon and Tandy Computers.

I have been using both makes for some years and apart from a very small amount of component failures (1 modulator, 1 joystick input chip), I have had no other problems.

The main trouble seems to be that your readers only like to play games. The Dragon and Tandys are amongst the most powerful of the 8 bit computers I have ever seen. I have been running random access sorts (several Megs long) and other business and control programs including using the *Telewriter* word processor programs.

I have four discs of 750K each on each Dragon, and have not even had a seek error on a run program in six years. I wish I could say the same of Tandy Dos (it does five or six seeks per read or write!).

The reason for this low error rate is that I run the so called obsolete *DeltaDos* which runs like a good mainframe unit. It allows chaining and other important features, and does not have any of the bugs that I keep reading about in your journal. I also run both Mx80 and Mx100 Epson printers and a Juki daisy wheel printer. The Juki has only two problems (it is slow and it throws away any characters over one line length, you cannot line wrap as with the Epsons). I feel it is time someone spoke up for these very high powered but small computers. By the way what has happened to Eurohard the company that bought out the Dragon? Are they still in business? If so can Dragons still be obtained from them (even if the ROMS have to be changed to English ones)?

I worked for many years on mainframe computers and I can run the same sort of jobs in the same sort of time on a Dragon! I have not seen it done on any other small computer.

A.C.Earnshaw
23 Dellfield Crescent
Cowley
Uxbridge
Middlesex
UB8 2ET

WELL, er, thanks for the tips, A.C. Actually, most of the folks I know who work on mainframes spend their lunch breaks playing 'Colossal Adventure'.

As far as we know, Eurohard are still in business, but no-one over here seems to be able to buy anything from them, so we aren't entirely clear what 'in business' consists of.

Footnote

Footnotes ... Don't forget to write to us if you would like to write about your disc drives for Dragon User ... at last Ken Smith got his Basic 42 letter printed — even if it did catch up with the next one.

News desk

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 4343

Dragon Show in September

PULSAR Software are organising a Dragon show in Rochdale, Lancashire in September.

The show, titled the North of England Dragon Show and Convention, will take place on Saturday 12th September at the Bishop Hanslow Upper School, Rochdale.

"We have been in touch with all the software dealers" says Pulsar, " and they all want to come, apart from Microdeal who weren't too sure. Its too early to finalise anything yet, but we have a panel of 'experts' from the local user group, and people to do demonstrations. We want to make this a bit more wide ranging than a normal show."

"There will be a computer clinic, where anyone can come and ask about any computing problems they are having, and we'll be having a forum where people can ask questions."

'We've organised a snack bar and we are applying for a bar licence." The show will be open from 10am to 4pm, and the gate charge will be £1.50 for adults and £100 for children.

Subject to the normal limitations imposed by doing three jobs at once, the editor of Dragon User is looking forward to finding out where Rochdale is in time for September.

New Trojan light pen software

MACGOWAN Consultants have rewritten the previously disappointing software which accompanied the Trojan Light Pen. The new version is available on tape for £7.00 or disc for £8.00. The Light Pen itself is also available for new users.

Among the improvements in the new software are full screen availability, all machine-code program , and PMODE 3 and 4 screen dumping for 114 different printers.

"All our products are printer orientated" says Bill MacGowan. "I am the country's leading expert on printers now, and as far as I am concerned, anybody can ring us up for advice on printers at any time with no obligation. I'd rather they did, because I can then tell them what's wrong with the printer they want to buy."

MacGowan, who is also writing screen dumps for Commodore, BBC and Atari micros, stresses that their products are unconditionally guaranteed and "are up to the standard which our customers have come to expect." MacGowan consultants are at 6 Arnhem Drive, Caythorpe, Nr. Grantham, Lincs NC32 3DQ (tel: 0400 72085), but their products can also be obtained from Harris Micros.

Printer progs

DRAGONFIRE Services have three new utility launches, all under £5.00. *Printer Prompt II* (version 2.5) is a new version of that program giving a 64 character-per-line screen display, WYSIWYG, double width letters and an 'address blocker' routine.

Versions are available for the Brother HR-5 TTP, Seikosha GP100A, and Epson FX compatible printers. Custom versions can be supplied — SAE for details. Price: £4.00.

The *Seikosha Screen Dumper* does just that, dumping hi-res modes 0, 2 and 4 in mini or large (roughly A4) size to the Seikosha GP100A printer. Price: £2.00

Fontasia Computa-Text is a program for use with MacGowan Consultants' *Printer Control* word processor. It gives a new text font in a squared, semi-graphic style, useful for display or for short texts like invitations or messages. Price: £2.00.

These programs can be obtained from Dragonfire Services, 13 Parry Jones Close, Blaina, Gwent, NP3 3NH.

High adventure

COWEN Software called to say that *Adventure Writer*, reviewed in last month's DU, is now available only as an improved version at the higher price of £14.95, and is being distributed by Peaksoft.

Modem package

A complete modem package for the Dragon 64 is being marketed by Hayton Electronics for £45, including post and packing.

The package includes a Prism Modem 2000, a connecting lead, software to enable the computer to access viewdata type boards (Prestel, for example) and scrolling ASCII boards which run on 100/75 baud, a list of over 200 bulletin boards countrywide, and full instructions.

The software comes on tape or disc for DragonDOS only. Hayton Electronics is at 36 Laurel Drive, Willaston, South Wirral, L64 1TW.

South coast show

PHILIP Beed, Dragon person and NDUG regular, is putting around the word that he and some south coast friends are organising a non-commercial 'show and get-together' in the late autumn or early winter. The likely venue will be in Gosport or Portsmouth, and chosen to suit numbers.

Philip says that his intention is primarily to get people together to discuss and demonstrate aspects of computing, but that if sufficient numbers are interested they may be able to organise discounts on some computer products. The cost of mounting the meeting should, he reckons, be between £1 and £3 per head.

All those interested in taking part or going along should contact Philip at 27 Findon Road, Elson, Gosport, Hants PO12 4EP with a stamped self addressed envelope for the latest update.

67 Dragons wedged in a letterbox

TALKING of Lancashire, computers — other computers, that is — are not your Editor's favourite subject just at the moment. Nor are they the favourite subject of reader P. Clayton of Chorley, Lancs.

The rest of us blame the computer when *Dragon User* doesn't arrive on time. Mr. Clayton is blaming the computer because *Dragon User* did turn up on time — all 67 of them.

In fact, at the time of writing, Mr. Clayton has received 63 and is awaiting the final four.

"I couldn't believe my eyes when the postman brought the sack round." he said.

Comments office supremo Anne Marie Allen: "The bug must have been waiting since his subscription was entered a year ago. Fortunately, he hasn't received 67 copies every month."

So if your *Dragon User* was a little late this month, check the postmark — it might have been to Lancashire.

News desk

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 4343

New company sells fifth generation

METASOFT is a new software house launched by Chris Jolly, who is already well known for his work on Dragon software.

Metasoft's first production is *Prolog*, which runs under OS-9 on a Dragon 64.

It is 'an extremely powerful 5th generation language for applications demanding built-in intelligence.'

Prolog programs consist of facts, rules and relationships, and execution consists of a controlled deduction through the facts and rules to arrive at a conclusion.

Says programmer Alain Colmerauer "the programmer who comes to Prolog after using a classical language will experience the same sort of revelation as someone who moves from arithmetic to the first steps of algebra."

Prolog costs £29.95, including post and packing and an 100 page manual, from Metasoft at 4 Pinehurst Walk, Orpington, Kent BR6 8DD.

Reprocess

DR. Anthony Daniels writes to say that there is a small bug in last month's *Graphics Screen Word Processor.*

The correct line 570 is:

```
570 IF PEEK(341)=247 AND
LS<>B THEN AB=B:B=LS:
S=LS*512:GOSUB3390:
IF AB>0 THEN MX=A:
MY=LS*512:GOSUB3920:
X=Y:POKEMY+XA,143::
RE=0:GOSUB260: B=AB:
Y=480:M=143:XA=X:AB=0:
PRINT@X,CHR$(143);:
PRINT@0,CHR$(143):
GOTO500 ELSE 500
```

Data protection survey reveals public approval

A survey commissioned by the Data Protection Registrar shows that seven out of ten people surveyed wanted to be able to gain access to information about themselves held on computer files. There was also support for companies charging a small fee — up to £5 — for supplying data on request.

Most interestingly, a majority of those asked wanted to see legislation to control the type of data which can be kept, to establish the right to have personal data added to or removed from a file where relevant, and to extend these protections to paper records as well as magnetic records. The most important right was felt to be that of inspecting records, and having them corrected where necessary.

Personal privacy as a matter of concern was ranked fourth after law and order, unemployment and education. People were most concerned about the misuse of financial information and medical records.

Surprisingly perhaps considering the publicity which is given to these issues from time to time, there was a high level of trust in the police, health service, and banks, while mail order companies and credit reference agencies were regarded with more suspicion.

While we must assume that the DPR's office want to show a level of public support for the Act after the consternation that it has caused among computer data users, there is no doubt that people *are* concerned that information about them might be stored and used without any means of cross checking its accuracy.

If the Data Protection Act can prevent this to any degree, it will be doing a good job.

Dragonsoft

New software for review should be sent to *Dragon User* 12-13 Little Newport Street, London WC2H 7PP.

Great modem, shame about the software

Package: *Dragon 64 Prism Modem 2000 comms package.*
Supplier: Hayton Electronics, 36 Laurel Drive, Willaston, S. Wirral L64 1TW.
Price: £45 including post and package. Disc or tape version available.

WHEN a complete communications package, leads and all, is on offer at this price, the first thing you ask yourself is "Where's the catch?". Well, I have to say that after two months, I have not yet found one.

The modem is a very neat Prism 2000, which operates in 1200/75 (V23) mode or, for one way communication, 1200 baud half duplex.

All the Input/Output sockets are at the rear leaving the front clear for the three LEDs and the 'mode' and 'data' buttons. The three indicators are for Mains On, Line (to indicate that it is online), and Signal, which shows when a carrier wave is established and information can be exchanged.

Using the system could not be easier. Once the software has loaded, it is simply a matter of dialling the relevant number, using an ordinary phone plugged into the back of the modem. As soon as the high pitched carrier tone is received, press the data button and replace the telephone receiver. You should then be online, and receiving data from the host computer. The telephone can be left plugged in and will still function normally when the computer is not using the line. The hardware works very well; it copes admirably with my noisy phone line without corrupting the incoming data. In fact the only time this has occurred was when someone tried to use the extension.

The software consists of four programs on a cassette and is the weak point of the whole package. The instructions can best be described as vague and the programs themselves have an amateur feel to them.

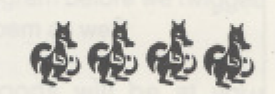
One example of this is in the VIEWDATA program. If you are not going to provide a save to tape routine, and I agree it is impractical, then there should be a DOS detection routine to stop the system crashing when a tape user presses the wrong key. Likewise, in the bulletin board program, what excuse can there be for allowing the buffer to overflow and corrupt the program? A 'buffer full' routine which referred the user back to the main menu could have saved a lot of aggro.

Even the MENU program has a problem; it lacks a CLEAR command. Any attempt to read back the information stored on disc without first running the ASCII routine results in a system crash. There is also no provision for reviewing saved VIEWDATA screens. These may be fairly insignificant faults to an experienced programmer, but to anyone relying solely on the software supplied, it could be enough to make them give up altogether.

On the plus side, Tim Hayton is always ready to enter into correspondence to sort out your problems and, so far as backup is concerned, you will not find better. He moved heaven and earth to find me a POKE to cure the problems I experienced with my SUPERDOS, and helped me out with finding local boards.

Taken overall, the package represents outstanding value for money, the price really is unbelievable. If only the software had been a little more professional, it would have been truly superb.

Ken G. Smith



The Answer

This is Gordon Lee's own solution to the May competition see page 31 for results

ANSWER: The best result that I could find using familiar words is the four word sequence GOD, OWL, SAP, and WET. Other possibles would include ADD, BEE, ILL, and LOO (its a card game!), or AHA, (a sunken fence), BIB, NUN, and TAT (a type of coarse canvas).

The program is built on a data-base containing 553 three-letter words. These were typed in from a suitable dictionary — in this case Cassell's *Crossword Finisher.* Of the 700 or so three-letter words listed, I did not include the less common words. These words are placed in the DATA lines (250 to 640).

The program takes each word in turn and then produces a series of twenty-six possible words by advancing each of the letters one place at a time. Each of these three-letter combinations is then checked against the list of words held, and any that match are held in the array R$. If the number held in this array during a single run exceeds 2, then the contents of the array are printed out (lines 200 and 210).

In order to speed up the running time of the program a number of time-savers are incorporated. Lines 130 to 150 check each set of letters to see if at least one vowel (A,E,I,O,U, or Y) is present. Time is saved by only checking a small section of the array relevent to the initial letter of the word.

This initial letter is denoted by variable A (line 70), and this value is checked against the array S() which holds the relative positions of words beginning with the letter L (ASCII code 76) will be found in the array between W$(278) and W$(304). This routine is at lines 170 to 190. Finally, line 180 also removes from the array each word as it is found to prevent re-duplication of sequences.

```
10 W=553
20 DIM W$(W),R$(10),S(27)
30 FOR N=1 TO W:READ W$(N):NEXT
40 FOR N=1 TO 27:READ S(N):NEXT
50 FOR N=1 TO W
60 R=1:Z$=W$(N):R$(R)=Z$:IF Z$=""THEN 220
70 A=ASC(MID$(Z$,1,1)):B=ASC(MID$(Z$,2,1)):C=ASC(MID$
(Z$,3,1))
80 A=A+1:IF A=91 THEN A=65
90 B=B+1:IF B=91 THEN B=65
100 C=C+1:IF C=91 THEN C=65
110 Z$=CHR$(A)+CHR$(B)+CHR$(C)
120 IF Z$=R$(1) THEN 200
130 FL=0:FOR F=1 TO 3:L$+MID$(Z$,F,1)
140 IF L$="A"OR L$="E"OR L$+"I"OR L$="O" OR L$="V"OR
L$="Y" THEN FL=1
150 NEXT F
160 IF FL=0 THEN 70
170 MIN=S(A-64):MAX=S(A-63)
180 FOR F=MIN TO MAX:IF W$(F)=Z$ THEN
R=R+1:R$(R)=Z$:W$(F)="":F=MAX
190 NEXT F:GOTO 70
200 IF R<3 THEN 220
210 FOR F=1 TO R:PRINTR$(F);" ";:NEXT:PRINT
220 NEXT N
230 PRINT" TASK COMPLETE"
240 END
250 DATA ACE,ACT,ADD,ADO,AFT,AGE,AGO,AHA,AID,AIL,AIM,AIR,AIT,ALB
260 DATA ALE,ALL,ALP,AND,ANT,ANY,APE,APT,ARC,ARE,ARK,ARM,ART,ASH
270 DATA ASK,ASP,ASS,ATE,AUK,AVE,AWE,AWL,AWN,AXE,AYE,AYR,BAD,BAG
280 DATA BAN,BAP,BAR,BAT,BAY,BED,BEE,BEG,BEL,BEN,BET,BIB,BID,BIG
290 DATA BIN,BIT,BOA,BOB,BOG,BOO,BOT,BOW,BOX,BOY,BRA,BUD,BUG,BUM
300 DATA BUN,BUS,BUT,BUY,BYE,CAB,CAD,CAM,CAN,CAP,CAR,CAT,CAW,CHI
310 DATA COB,COD,COG,CON,COO,COP,COR,COS,COT,COW,COX,COY,CRY,CUB
320 DATA CUD,CUE,CUP,CUR,CUT,DAB,DAD,DAM,DAN,DAW,DAY,DEB,DEN,DEW
330 DATA DID,DIE,DIG,DIM,DIN,DIP,DOE,DOG,DOH,DON,DOT,DRY,DUB,DUD
340 DATA DUE,DUG,DUN,DUO,DYE,EAR,EAT,EBB,EEL,EGG,EGO,EKE,ELF,ELK
350 DATA ELL,ELM,EMU,END,EON,ERA,ERE,ERG,ERR,EVE,EWE,EYE,FAD,FAG
360 DATA FAN,FAR,FAT,FAY,FED,FEE,FEN,FEW,FEY,FEZ,FIB,FIE,FIG,GIN
370 DATA FIR,FIT,FIX,FLU,FLY,FOR,FOE,FOG,FOP,FOR,FOX,FRY,FUG,FUN
380 DATA FUR,GAB,GAD,GAG,GAL,GAP,GAS,GAY,GEE,GEL,GEM,GEN,GET,GIB
390 DATA GIG,GIN,GNU,GOB,GOD,GOG,GOO,GOT,GUM,GUN,GUT,GUY,GYM,GYP
400 DATA HAD,HAG,HAH,HAM,HAP,HAS,HAT,HAW,HAY,HEN,HER,HEW,HEX,HEY
410 DATA HID,HIM,HIP,HIS,HIT,HOB,HOD,HOE,HOG,HOP,HOT,HOW,HUB,HUE
420 DATA HUG,HUM,HUT,ICE,ICY,IDE,ILK,ILL,IMP,INK,INN,ION,IRE,IRK
430 DATA IVY,JAB,JAG,JAM,JAP,JAR,JAW,JAY,JET,JEW,JIB,JIG,JOB,JOG
440 DATA JOT,JOY,JUG,JUT,KEG,KEN,KEY,KID,KIN,KIP,KIT,LAB,LAD,LAG
450 DATA LAH,LAM,LAP,LAW,LAX,LAY,LEA,LED,LEE,LEG,LET,LID,LIE,LIP
460 DATA LIT,LOB,LOG,LOO,LOP,LOT,LOW,LUG,LYE,MAD,MAG,MAN,MAP,MAR
470 DATA MAT,MAW,MAX,MAY,MEN,MET,MEW,MID,MIX,MOB,MOD,MOO,MOP,MOW
480 DATA MUD,MUG,MUM,NAB,NAG,NAP,NAY,NEE,NET,NEW,NIB,NIL,NIP,NIT
490 DATA NIX,NOB,NOD,NOG,NOR,NOT,NOW,NUB,NUN,NUT,OAF,OAK,OAR,OAT
500 DATA OBI,ODD,ODE,OFF,OFT,OHM,OIL,OLD,ONE,OPT,ORB,ORC,ORE,OUR
510 DATA OUT,OVA,OWE,OWL,OWN,PAD,PAL,PAN,PAP,PAR,PAS,PAT,PAW,PAX
520 DATA PAY,PEA,PEG,PEN,PEP,PER,PET,PEW,PIE,PIG,PIN,PIP,PIT,PLY
530 DATA POD,POP,POT,POX,PRY,PUB,PUD,PUG,PUN,PUP,PUS,PUT,RAG,RAJ
540 DATA RAM,RAN,RAP,RAT,RAW,RAY,RED,REF,REP,REV,RIB,RID,RIG,RIM
550 DATA RIP,ROB,ROC,ROD,ROE,ROT,ROW,RUB,RUE,RUG,RUM,RUN,RUT,RYE
560 DATA SAC,SAD,SAG,SAP,SAT,SAW,SAY,SEA,SEE,SET,SEW,SEX,SHE,SHY
570 DATA SIN,SIP,SIR,SIT,SIX,SKI,SKY,SLY,SOB,SOD,SON,SOP,SOT,SOU
580 DATA SOW,SPA,SPY,STY,SUB,SUE,SUM,SUN,SUP,TAB,TAG,TAN,TAP,TAR
590 DATA TAT,TAW,TAX,TEA,TEE,TEN,THE,THY,TIC,TIE,TIG,TIN,TIP,TIT
600 DATA TOD,TOE,TON,TOO,TOP,TOR,TOT,TOW,TOY,TRY,TUB,TUG,TUN,TUT
610 DATA TWO,URN,USE,VAN,VAT,VET,VEX,VIA,VIE,VIM,VIZ,VOW,WAD,WAG
620 DATA WAN,WAR,WAS,WAX,WAY,WEB,WED,WEE,WEN,WET,WEY,WHO,WHY,WIG
630 DATA WIN,WIT,WOE,WON,WOO,WOW,WRY,YAK,YAM,YAP,YAW,YEA,YEN,YES
640 DATA YET,YEW,YON,YOU,ZEN,ZIP,ZOO
650 DATA 1,41,76,104,132,153,184,211,242,254,271,278,304,326,347
660 DATA 370,405,405,435,472,506,508,517,540,540,551,553
```

The April Answer

Remember how we couldn't fit last month's Answer in? Well, here it is. Along with some of their poetry . . .

Sector	Final Score (X)	Cube root of X
20	4096	16
1	1	1
18	15625	25
4	1728	12
13	512	8
6	216	6
10	27	3
15	343	7
2	64	4
17	1000	10
3	64	4
19	125000	50
7	125	5
16	5832	18
8	8	2
11	1000	10
14	2744	14
9	9261	21
12	4096	16
5	343	7

Solution

The final totals can be obtained by repeatedly summing up the scores as described, checking the running total at each stage to see if it is a perfect cube. However, for the larger scores this could be a very lengthy process. A faster result can be achieved if it is realised that on the first circuit of the board, from each starting point, there is a series of twenty ascending subtotals. For example, from sector 20 the series would run: 20, 21, 39, 43, 56, . . . and so on up to a total of 210 after a full circuit of the board. Each subsequent circuit of the board would result in running totals of these scores plus successive multiples of 210.

Therefore it is faster, and mathematically a lot safer, to take an ascending series of cubes and test each one to determine if it is a possible target score in the series

$$210n + R$$

where n is any integer (including zero) and R is one of the subtotals for each starting score.

In the program, the array sector holds the 20 scores from the dartboard in clockwise sequence. These are read from the DATA lines. The loop DART takes each of the twenty starting scores in turn and computes the twenty subtotals from the first circuit of the board. These are held in array RESIDUE (lines 110 — 150).

An ascending series of cubes is then tested to determine if it could be apossible score. This is done by dividing the cube by 210 and taking the remainder. If this remainder is one of the residues in the array then a possible result has been found. In practice, the remainder is found by repeatedly subtracting 210 from the number under test.

```
10 DIM SECTOR(20),RESIDUE(20)
20 RESTORE
30 FOR N=1 TO 20:READ X:SECTOR
   (N)=X:NEXT
100 FOR DART=1 TO 20
110 T=0:F=DART
120 T=T+SECTOR(F)
130 RESIDUE(F)=T
140 F=F+1:IF F=21 THEN F=1
150 IF T=210 THEN 200 ELSE 120
200 N=1:FL=0
210 C=N*N*N
220 R=C
230 IF R<=210 THEN 250
240 R=R-210:GOTO 230
250 FOR F=1 TO 20
260 IF RESIDUE(F)=R THEN PRINT
    SECTOR(DART),C,N:FL=1
270 NEXT F
280 IF FL=1 THEN 500
290 N=N+1:GOTO 210
500 NEXT DART
1000 DATA 20,1,18,4,13,6,10,15,2,17
1010 DATA 3,19,7,16,8,11,14,9,12,5
```

The winners (and non-winners) of the April Competition came up with some rilly amazing poems. Barely daunted by the fact that they had to rhyme with 'tangle', out came the Thesauri (a distant relation of the Dragon) and the pentameters.

Ode to despair

MY wires are in a tangle
My dongle will not dangle
And life's one constant wrangle
While I'm looking for the angle
To this competition jangle.

Alan Thomas

Keep it short

A long man am I
And stuck in a tangle.
Better perhaps
Than stuck in a mangle!

Richard Long

Epic

MY poor old Dragon's all confused
It's really in a tangle
It's full of bits, the keyboard fits
At such a funny angle.

The stacks of ribbon cables too
They'd make a lovely bangle
With help from Sam, I'd love to ram
The buffers as they dangle.

The bus has lost control it seems
It makes an awful jangle.
But pointers from an 8K ROM
Will sort out any wrangle.

The chips are dry, they ought to be —
I've put them through a mangle
Which also wrecks my faith in HEX
When flattened like rectangles.

My Dragon's roar is muted now
I'll use my last try-angle
For me the grog, for it the prog.
That's it! My greatest wangle.

Fred Willers

Criss Cross Crash

There was a young Scot from Kirkangle
Whose braces got caught in a tangle.
He said with a frown
As his new kilt fell down
"I guess it will just have to dangle."

Terry Potter, who swears he meant the sporran.

Has bin

A Dragon I thought with a grin
Will help me these puzzles to win.
But their logical tangle
Just makes my brain jangle.
Well, just one more try and then — bin.

M. Armitstead

Dreadlock

Let us strangle with a bangle
Or entangle with a mangle
Those who coerse from us such verse
Deserve this curse or something werse.

P.D.Maddocks, who ran his poem straight on from his solution, and nearly lost points for writing such a long program before we twigged that it generated the poem as well!

Gentlemen, the Vogons will be at your doors in a week.

Puzzling over FORTH

J. B. Slinger investigates the powers of the FORTH language by untangling some past Gordon Lee puzzles

WHEN I heard that Mr Cadge was writing an article on the Forth language, I thought that it might interest some readers to read about my experience with the language. I add that I only have a limited experience and thus my viewpoint is not far removed from that of a reader who might be wondering whether to take up the language.

I bought a cassette version and as such it does not have the facilities of the full Forth system but it has given me a great deal of fun. Accounts of Forth usually pay great attention to the use of the STACK, often to the extent of failing to mention that there are facilities to store values in variables and constants as in Basic. It is, in fact, quite easy to translate a Basic program directly into Forth and this is how I started.

I had to remember that Forth uses integer arithmetic: at first this was a frightening concept, but it has not turned out to be a real problem. In fact, it has often been beneficial as I will illustrate in the examples below but before doing so I will return to a brief comment on the use of the STACK.

In my (limited) experience, most of the fun in using Forth comes from careful planning to use the STACK to the limit of one's nerve at the time, making judicious use of variables to simplify stack manipulation.

Systematic paperwork is crucial. I favour working in columns on A3 paper. In the left hand column I write the stack contents, with the most accessible item to the right (that is to say nearest to the centre of the page). Incidentally, it is conventional to describe the most accessible item as being on 'top' and I shall do so. In the centre column, in the next line, I write the operating instructions. In the left hand column of the third line, I write the stack contents after the instructions have been carried out. The effect is that each group of instructions has the 'before' and 'after' stack contents shown above and below it. If there are RETURN STACK instructions (see below), they are written in the right hand column with the 'top' item nearest to the centre of the page; most often the Return Stack is used as a temporary store for items from the normal stack so this orientation is logical in use. It is as if the items flip across the centre of the page. Alternatively the third column is used for remarks. The examples that I have chosen are from Mr Lee's puzzle page.

June puzzle

This was to find a four-figure number which repeated itself after four cycles: in each cycle, the number has to be squared and the central four digits have to be isolated for the next cycle. The key routine is that which isolates the central digits. In Basic it can be done with 'string' functions or by division and taking the integer parts of the quotient or remainder as appropriate. Forth arithmetic is simpler because it is, by definition, in integer form. Note, the square has to be held as a 'double number' (or a 32-bit number) because it is too big for a normal 16-bit Forth number. For this reason I used the operator U* which multiplies two single length numbers to give a double length one. The procedure for manipulating the square is as follows:

CYCLE: The purpose is as described. It needs an initial single length number on the stack and finishes with another single length number in its place.

STACK	OPERATION	REMARKS
seed		initial value
	: CYCLE	beginning
seed		no change
	DUP U*	duplicate & square
square		double length
	100 M/MOD	divides by 100
remainder & quotient		single length double length
	ROT DROP	drop remainder
quotient		double number
	10000 M/MOD	div. by 10000
remainder & quotient		
	2DROP	drop quotient
remainder		new seed
	;	end of routine

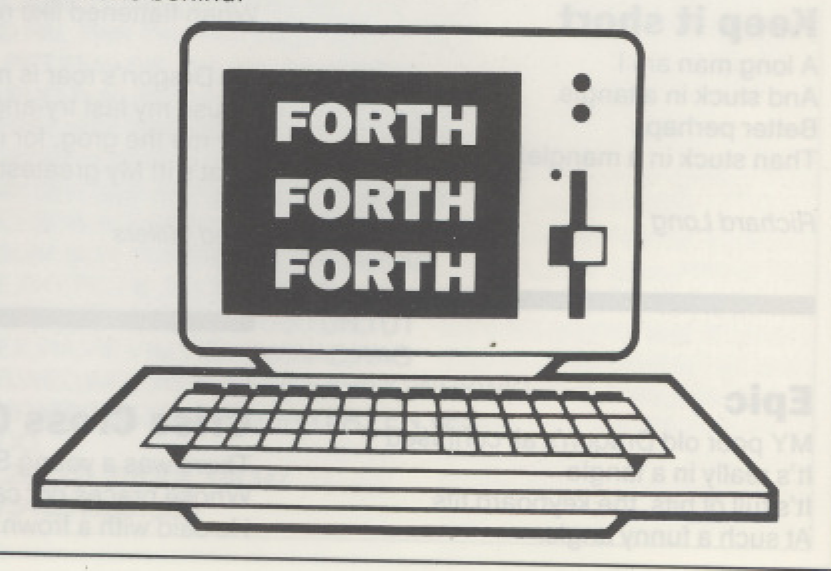
As will be shown below CYCLE is part of a larger routine. In that sense CYCLE is a sub-routine but it is also a complete program. The authors of Forth recognised that there was a problem of nomenclature; they kept their options open by coining the name WORD to refer to such mini-programs.

Before moving on to talk about the larger routine, it is prudent to check that CYCLE does do what it is supposed to do. There are two things to check. The first is whether it does, in fact, isolate the central four digits of a square. The second may be unexpected to a Basic programmer but it is vital to Forth: does the routine leave stray numbers on the stack, or does it steal numbers off the stack? If either happens, it is a disaster.

The way to check CYCLE is to type a number, a space, the word CYCLE, a space, a dot (which means print a number), and then ENTER. The computer should display the answer and its accuracy can be checked for a whole range of numbers. It is necessary to check two and three digit numbers because such can arise (because of leading zeros) in the main routine. The stack effect can be checked by typing a dummy number before the seed, repeating the test, and then typing another dot; all being well the computer should display the dummy number.

Assuming that all is well with CYCLE, the main word JUNE-PUZZLE can be developed as follows:

JUNE-PUZZLE: Print numbers which are repeated after 4 CYCLES. It does not need a number on the stack and does not leave one behind.



Note:
The article by Brian Cadge, Dragon FORTH, appeared in DU February 1987. Brian Cadge was using NDUG FORTH, available from the National Dragon User Group, 6 Navarino Road, Worthing, Sussex. Tel. Worthing 207585. J.B. Slinger is using Teleforth *on cassette from Microdeal, on a Dragon 32 without the disc controller in place. A page of programs used in the three examples is printed at the end of the article.*

STACK	OPERATION	REMARKS
empty		
	: JUNE-PUZZLE	start
empty		
	10000 1000 DO	(start of loop
		(like FOR I =
		(1000 to 9999
empty		
	III	(3 copies of
counter test seed		(loop counter
	CYCLE CYCLE	see above
	CYCLE CYCLE	
counter test result		
	=	(tests if 2 top
counter flag		(numbers equal
	IF."YES"CR	
empty		(print answer
	ELSE DROP	(else drop
empty		(counter
	THEN	(end of it
	LOOP	(like NEXT-STEP
empty		
	;	end of word

The "word" takes about five to twelve minutes. It prints thirteen solutions, but twelve can be eliminated by visual inspection using some additional information given in the original article. The routine could be improved by using part of the AUGUST-PUZZLE routine instead of the DO LOOP. As a final comment, it is easy to see a similarity with Basic but the Forth CYCLE is much neater than a Basic sub-routine would be. There is very little stack manipulation because very little is needed. There is more in the next example.

July puzzle

This involves division to 4000 odd decimal places. Each successive digit in the quotient is used as a direction indicator for a 'step'. At first sight, the integer arithmetic of Forth would appear to be at a disadvantage compared with the floating point arithmetic of Basic but the reverse is true. With Basic, one has to take great care to avoid rounding-off errors by using a sequence such as :

DIGIT = INTEGER of DIVISION
REMAINDER = DIVIDEND- DIGIT*DIVISOR
NEW-DIVIDEND = INTEGER of 10*REMAINDER

In Forth, the sequence

OLD-REMAINDER 10 17389 */MOD

gives the new remainder and the digit directly onto the stack. The digit is on 'top' and available for determining the movement. The procedure is long division.

A way to describe the movement is to think of a graph. A horizontal (East-West) movement is a change in the X-coordinate, a vertical (north-south) movement is a change in the Y-coordinate, and a diagonal movement is a change in both.

I imagined that the initial coordinates were 0,0 and tested to see if either had reached 50 after each move. Note that the coordinates were not variables as in Basic, they were simply numbers on the stack distinguished only on my piece of paper by their relative position on the stack.

In designing the word, the most important decision was the order of the various numbers on the stack. The procedure involved moving top numbers from the normal stack to the Return Stack to gain access to less accessible numbers as in the game *Towers of Hanoi*; this is said to be much quicker than putting numbers into storage as variables.

The symbols for transfer to and from the Return Stack are ⟩R and R⟩ (imagine arrows). There is also an instruction to make a copy of the top numbers from the Return Stack to the top of the stack ; it is R in my version of Forth but I believe that R@ is more usual.

The consequence of this is that the part of the word which has the inter-stack transfers is difficult to break-up into small pieces for testing purposes. So another technique has to be used; this involves use of the "Screens". These are a feature of Forth whereby it is possible to type a word onto a screen where it held as typed in text form. The contents can be 'loaded' (or compiled), and tested. Afterwards the screen's contents can be recalled and edited as necessary. This is a most useful feature. In the present instance, one could omit the loop structure and the long division initially. The rest could be typed in the centre of the screen, and checked, then the remainder could be added and the whole loaded and checked.

The word is in six parts. The first part merely puts a seed remainder and 3 zeros onto the stack. From the top, the zeros are the X-coordinate, the Y-coordinate and an optional counter. The second, third, and fourth parts of the word are in a loop and are repeated until one of the coordinates reaches an absolute value of 50. In the second part, the coordinates are shifted onto the Return Stack. The counter is incremented and shifted across also. These actions expose the old remainder as the top element of the stack. The long division is the third part of the word. It leaves a new remainder and a quotient on the stack. The latter is on top and handy for determining the direction of the step.

The next major part of the word is a whole series of IF ELSE THEN tests. It is necessary to make a duplicate copy of the quotient for each test: when a match is found, the original quotient is dropped as no longer required. Depending on the value of the quotient, the counter and the coordinates are brought back onto the stack and changed as required.

It is usual to show a series of IF ELSE THEN clauses in a indented form so that one can check which ELSE matches which THEN.

However in this case I decided to use a different presentation to make it easier to see how the routine works and, more important, to make it easier to check the stack effect. Each route must leave the Return Stack empty, and the stack with, from the top, X-coordinate, Y-coordinate, counter, new remainder.

In the fifth part, the two coordinates are duplicated, 'OVER OVER', and the copies are tested to see if either has reached the critical value. If so the result is printed, if not the calculation is repeated. The presentation of the result is not as specified in the *Dragon User* but can be converted into the correct form.

JULY-PUZZLE: The routine does not require an initial value to be put on the stack, and does not leave one behind.

STACK	OPERATION	RETURN STACK (& remarks)
empty		
	: JULY-PUZZLE	(beginning)
	10000 0 0 0	(values)
seed counter Y-coord X-coord		
	BEGIN	(begin loop)
no change		
	⟩R ⟩R 1 + ⟩R	COUNTER Y- X-COORDS
seed = remainder		
	10 17389 */MOD	(long div'n)
remainder quotient		(no change) (of RET. STACK)

```
DUP 1 = IF DROP R⟩ R⟩ 1 + R⟩
ELSE DUP 2 = IF DROP R⟩ R⟩ 1 + R⟩1+
ELSE DUP 3 = IF DROP R⟩ R⟩ R⟩ 1+
ELSE DUP 4 = IF DROP R⟩ R⟩ 1 − R⟩1+
ELSE DUP 5 = IF DROP R⟩ R⟩ 1 − R⟩
ELSE DUP 6 = IF DROP R⟩ R⟩ 1 −R⟩1-
ELSE DUP 7 = IF DROP R⟩ R⟩ R⟩ 1-
ELSE DUP 8 = IF DROP R⟩ R⟩ 1 + R⟩1-
ELSE DROP R⟩ R⟩ R⟩
THEN THEN THEN THEN
THEN THEN THEN THEN
```

STACK	OPERATION	RETURN STACK (& remarks)
remainder counter Y-coord X-coord		(empty)

STACK	OPERATION	RETURN STACK
	OVER OVER	(see text)
as above + copies of coords		
	ABS 50 =	(test X-coord)
as above except flag instead of copy of X-coord		
	SWAP	(copy of (Y-coord top)
	ABS 50 =	(test Y-coord)
remainder counter Y-coord X-coord flag flag		
	OR	(either true?)
remainder counter Y-coord X-coord flag		
	UNTIL	(end of loop) (structure)
remainder counter Y-coord X-coord		
	. . . .	(print out)
empty		
	;	(end of word)

Before leaving the description of the JULY-PUZZLE, it is worth explaining what '*/MOD' does. It multiplies two single length numbers to a double length number which is then divided by a third single length number; there would be an overflow error with the combination of '*' and '/MOD' as this only has a single length intermediate product (10 times 10000 would be too big for it).

August puzzle

This is a much bigger problem altogether. The idea is that the words DRAGON and USER represent two numbers. Each different letter is a different digit.

We are told that if the one is divided by the other, it will give a two figure integer quotient. The cube of this will produce an English word if its digits are substituted by the letters used in the original code words.

Conceptually there are two ways to proceed. Either one can synthesise all possible DRAGON and corresponding USER words and check whether the product meets the criterion of different digits.

Although the first is easier to understand, it actually involves five times more combinations and hence takes five times longer. So I will describe the other method.

The key part is how to indicate whether a particular digit is being used or not. In Basic, one could use an array to store flags. The same is true of Forth, but one has to define the array first. In English, an array could be defined as a group of storage addresses starting at a known address. This means that any location can be described by quoting an index number from the starting address. In my version of Forth such a structure for 10 locations, can be defined by:

```
0 VARIABLE NAME 8 ALLOT
```

VARIABLE reserves storage for two bytes and 8 ALLOT reserves storage for eight more. The start address is given by "NAME". So 8 NAME + or NAME 8 + is the address of the ninth storage; more correctly it is a number which equals the address but it will be used as an address in this article. NAME could be almost anything, but I use ZERO because ZERO 8 + is suggestive of the way in which it will be seen to be used.

The storages in the array are used to store flags to show whether the corresponding digit is being used. I use the number 45 to show vacant, as this is the ASCII code for "—", and the ASCII code for the letter equivalent to the digit for 'occupied'. Thus if U is equivalent to 4, then the address ZERO 4 + contains 85. The advantage of this system is that Forth has a most useful word EMIT which prints the letter equivalent to an ASCII code. Thus

```
4 ZERO + C@ EMIT
```

would print U. C@ is the instruction to fetch (read) a single byte from an address. Jumping ahead to the ultimate decoding of the cube numbers, this is done by breaking the number into a list of digits and using each one to EMIT the corresponding letter. It will be noticed that I use ASCII codes for upper case U S E and R and lower case codes for the other letters.

This is because there are many decode jobs to be done for each value of the USER number and it is easy to write a word PURGE to remove the lower case codes from the array without affecting the upper case codes.

Two other general points need to be explained. First it is pointless to multiply a USER number by a 2 figure number which is too small to give a 6 figure quotient ; the word LOW-LIMIT calculates a minimum multiplier.

The second point is that the most searching criterion for possible DRAGON numbers is to check whether the second digit is the same as the last digit of the USER number. By trial only 10 % of the possibilities pass this test so it is efficient to make it the first in the testing routine.

The word TEST begins with the USER number and the multiplier on top of the stack. They are copied and multiplied to a double length number (using U*) ; for the purposes of reference I call it DRAGON. It is first divided by 10000 to break it into a remainder AGON and a quotient DR which is then divided by 10. The remainder is checked against the true value of R which is recalled from the variable R-TRUE. It is a reverse check. IF the values are different, the test ends in failure. ELSE leads to further checks. In the event that any check after the first gives a satisfactory result, the appropriate storage receives the appropriate lower case ASCII code. If all checks are satisfactory, control passes to the word DECODE.

DECODE begins with a print-out of the multiplier, USER, DRAGON, and the cube. The cube is decoded as explained previously. A feature to note is that the order of the digits is reversed twice so that the end result is the right way round. Control then reverts to TEST.

The combination of TEST and DECODE makes sure that the array is PURGEd of lower case codes when necessary. The combined routine requires the USER number and the multiplier to be on the stack at the outset, and returns to the main routine without the multiplier. The overall stack effect is :-

other items. USER, multiplier --->
other items, USER

The other items come from the number selection routine.

The number selection routine is four nested DO LOOPs. The outer one selects the U digit and has first choice. It marks the array as explained. The other digits are selected one by one. In each case there is an IF test of whether the cell in the array is occupied (IF it is the number is rejected) ELSE the array cell is reserved and control is passed to the next DO-LOOP. It leaves a copy of the digit on the stack which is used for calculation purposes and for releasing the array cell on the outgoing side of the loop. For the sake of clarity I show the DO LOOPS separately but there is no objection to typing them in one structure.

After the R digit has been selected, the ELSE clause charges the variable R-TRUE, and calls the word CALC to calculate the value of the USER number. Then it calls the word LOW-LIMIT to calculate the minimum value of the multiplier.

The main bulk of the routine is done in a central DO-LOOP. The loop counter is placed on the stack and the word TEST is called. The workings of TEST and its dependant words PURGE and DECODE have already been described. The whole routine is summarised in a diagram using a sort of "pidgin English/Forth". Elements of each structure are directly underneath each other to show how they fit together. Such an overview is useful because the sub-words have to be typed first. Even before the sub-words, one defines the three variables as follows:

```
0 VARIABLE ZERO 8 ALLOT
0 VARIABLE R-TRUE
0 VARIABLE ? DIGITS
```

PURGE: There is no stack effect. The word replaces any value in the array which is greater than 96 by 45.

STACK	OPERATION	REMARKS
empty		
	: PURGE	start of word
empty		
	10 0 DO I	
counter		
	ZERO + DUP	calc address
addr' addr'		
	C@ 96	test contents of address
addr' flag		
	IF DROP	if lower drop
empty		
	ELSE 45 SWAP C!	else replace
empty		
	THEN LOOP ;	end

CALC: Calculates value of the USER number. Assumes stack contains four digits corresponding to U S E and R, with the last on top. The routine places the USER number on top of these. Makes use of the Return Stack for temporary holdings of S E R .

STACK	OPERATION	RETURN STACK (& remarks)
u s e r		(4 digits)
	: CALC	(start)
no change		
	>R >R >R	
u		s e r
	DUP 1000*	
u 1000*u		s e r
	R 100 * +	
u "u - -"		s e r
	R> SWAP	
u s "u s - -"		e r
	R 10 * +	
u s "u s e -"		e r
	R> SWAP	
u s e "u s e -"		r
	R +	
u s e "u s e r"	r	
	R> SWAP	
u s e r "USER"		(R/S empty)
	;	(end of world)

LOW-LIMIT: Calculates value of minimum multiplier (100,000/USER). Expects to find USER on top of the stack. Places result on top of it.

STACK	OPERATION	REMARKS
user		
	: LOW-LIMIT	start
no change		
	1000 OVER	to avoid double length multiplication
user 1000 user		
	100 SWAP	
user 1000	100 user	
	*/	(double length) (intermediate)
user result		
	;	end of word

DECODE: Expects to find 'user number' and 'multiplier' on the stack. Organises print-out of multiplier, user, dragon & cube. Decodes cube and prints as a word. Removes multiplier but leaves user number on the stack. Routine requires a variable ?DIGITS to store the number to digits in the cube (as calculated from the value of the multiplier).

STACK	OPERATION	REMARKS
user, multiplier		See above
	: DECODE	start
no change		
	DUP.	prints mult.
no change		
	OVER DUP.	prints user
user mult. user		
	OVER U* D.	prints dragon
user, mult.		
	DUP 22 < IF 4 ?DIGITS ! ELSE DUP 47 < IF 5 ?DIGITS ! ELSE 6 ?DIGITS ! THEN	mult. controls size of cube, noted in ?DIGITS THEN
user, mult		
	DUP DUP * U*	calcs. cube
user, cube		
	2DUP D.	prints cube
no change		
	?DIGITS @ 0 DO 10 M/MOD LOOP 2DROP	(divide by 10 (many times (drop quotient
user, digits		
	?DIGITS @ 0 DO ZERO + C@ EMIT LOOP CR	(no. digits (calc address (print word carrier ret.
user		
	;	end of word

TEST: Breaks a 6 digit number into individual digits and checks if already used. If so the routine ends, else it calls DECODE. Where needed PURGES the array. Passes user & multiplier on the stack to DECODE. As noted above DECODE returns user only.

STACK	OPERATION	REMARKS
user,mult.		see above
	: TEST	start of word
no change		
	OVER OVER U*	calcs dragon
as above + dragon		
	10000 M/MOD DROP	(divides (16-bit quot.
user,mult,agon,dr		
	10/MOD SWAP	r to top
user,mult,agon,d,r		
	R-TRUE @ — ABS 0 >	(read value (& compares
user,mult,agon, d,flag		
	IF DROP DROP DROP	(failure
	ELSE ZERO + DUP C@ 50 >	(calc addr. (read value (occupied?
user,mult, agon,adr.,flag		
	IF DROP DROP DROP ELSE 100 SWAP C!	(failure (reserve
user,mult,agon		
	10/MOD SWAP	n to top
user,mult,ago,n		
	ZERO + DUP C@ 50 >	(calc addr (occupied?

STACK	OPERATION	REMARKS
user,mult,ago,addr,flag		
	IF DROP DROP	
	DROP PURGE	
	ELSE 110 SWAP C!	(reserve
user,mult,ago		
	10/MOD SWAP	o on top
user,mult,ag,o		
	ZERO + DUP	(calc addr
	C@ 50〉	(occupied?
user,mult,ag,addr,flag		
	IF DROP DROP	
	DROP PURGE	
	ELSE 111 SWAP C!	(reserve
user,mult,ag		
	10 /MOD SWAP	g on top
user,mult,a,g		
	ZERO + DUP	(calc addr
	C@ 50〉	(occupied?
user,mult,a,addr,flag		
	IF DROP DROP	
	DROP PURGE	
	ELSE 103 SWAP C!	(reserve
user,mult,a		
	10/MOD DROP	(a on top
	ZERO + DUP	(calc addr
	C@ 50 〉	(occupied?
user,mult,addr,flag		
	IF DROP DROP	
	PURGE	
	ELSE 97 SWAP C!	(reserve
	DECODE PURGE	see text
	THEN THEN THEN	
	THEN THEN THEN	
	;	word end
user		

R-SELECT: Selects the R-digit, & places on stack. Uses it to calculate address and look at contents. If address occupied, digit is rejected, ELSE it is reserved by ASCII code for R. Charges variable R-TRUE ; calls CALC to calculate the 'USER' number. Calls LOW-LIMIT to calculate the minimum value of the multiplier. For every value of the multiplier, places the loop counter on the stack. At this point the top elements on the stack are a copy of the R-digit, the user number, and the loop counter. Then TEST is called for each value of the multiplier. Afterwards when the loop has been completed, the USER number is dropped ; the array address is released using the value of the R digit from the top of the stack.

STACK	OPERATION	REMARKS
u,s,e		initial
		values
	: R-SELECT	(start
	10 0 DO I I	(loop counters
	ZERO + DUP	(calc addr.
u,s,e,r, addr,addr		
	c@ 50 〉	occupied?
u,s,e,r, addr,flag		
	IF DROP DROP	reject it
	ELSE 82 SWAP C!	reserve it
u,s,e,r,		
	DUP R-TRUE !	fill variable
no change		
	CALC	user number
u,s,e,r,user		
	LOW-LIMIT	(limits
	100 SWAP	(for loop
u,s,e,r,user,100,low-limit		
	DO	inner loop
	I TEST	see TEST
	LOOP DROP	
u,s,e,r,		
	ZERO +	calc. addr,
u,s,e,addr		
	45 SWAP C!	free number

STACK	OPERATION	REMARKS
u,s,e,		
	THEN LOOP ;	end of word

E-SELECT: As R-SELECT picks next free digit and reserves it by storing 69 (ASCII code for E) in the array. Places value of digit on the stack; calls R-SELECT. Uses the value on the stack to release the array address. S-SELECT is identical except that it stores 83 in the array and calls E-SELECT.

STACK	OPERATION	REMARKS
u,s		already there
	: E-SELECT	start
	10 0 DO	(loop
	I I	counters
	ZERO + DUP C@	calc addr
	50〉	occupied?
u,s,e,addr,flag		
	IF DROP DROP	reject it
	ELSE 69 SWAP C!	reserve it
u,s,e		
	R-SELECT	see above
u,s,e		
	ZERO + 45	
	SWAP C!	release it
u,s,		
	THEN LOOP ;	end of word

U-SELECT: Reserves a digit between two and nine by storing 85 (ASCII code for U) in corresponding address of array. Places digit on stack. Calls S-SELECT. Afterwards uses the digit to release the address of the array. The overall stack effect is neutral.

STACK	OPERATION	REMARKS
empty		
	: U-SELECT	start of word
	10 2 DO I I	start loop
counter=u, counter		
	ZERO +	calc addr.
u,addr		
	85 SWAP C!	reserve
u		
	S-SELECT	see text
u,		
		ZERO + 45
SWAP C!		
empty		(free addr)
	LOOP ;	(end of word)

AUGUST-PUZZLE: Main word of routine. Empties the array and calls U-SELECT. All the routines have already been defined and call one another as required with the result that the final definition is very brief.

STACK	OPERATION	REMARKS
empty		
	: AUGUST-PUZZLE	
	ZERO 10 45 FILL CR	
		see above
empty		
	U-SELECT ;	end of word

The word takes about 50 minutes which compares very favourably with the running time of its BASIC counterpart. The latter took between three and four hours.

Concluding observation, I have had a great deal of fun planning the programs described above and I hope that I have interested one or two other readers to try FORTH. In writing the account I have drawn heavily on the Manual that came with my cassette and upon a book called *The Complete Forth* by Alan Winfield, published by Sigma Technical Press. I should like to thank the authors of the two sources.

```
: CYCLE DUP U* 100 M/MOD ROT
  DROP 10000 M/MOD 2DROP ;
: JUNE-PUZZLE 10000 1000 DO
  I I I CYCLE CYCLE CYCLE CYCLE
  = IF . ." YES " CR ELSE DROP
  THEN LOOP ;

: JULY-PUZZLE 10000 0 0 0 BEGIN
 >R >R 1+ >R 10 17389 */MOD
DUP 1 = IF DROP R> R> 1 + R>
        ELSE
DUP 2 = IF DROP R> R> 1 + R> 1 +
        ELSE
DUP 3 = IF DROP R> R>     R> 1 +
        ELSE
DUP 4 = IF DROP R> R> 1 - R> 1 +
       ELSE
DUP 5 = IF DROP R> R> 1 - R>
       ELSE
DUP 6 = IF DROP R> R> 1 - R> 1 -
       ELSE
DUP 7 = IF DROP R> R>     R> 1 -
       ELSE
DUP 8 = IF DROP R> R> 1 + R> 1 -
    ELSE  DROP R> R>     R>
THEN THEN THEN THEN THEN THEN
THEN THEN OVER OVER ABS 50 =
SWAP ABS 50 = OR UNTIL
 . . . . ;
                       2 LOAD ;S
 0 VARIABLE ZERO 8 ALLOT
 0 VARIABLE R-TRUE
 : PURGE 10 0 DO I ZERO + DUP
   C@ 96 < IF DROP ELSE 45 SWAP
   C! THEN LOOP ;
 : CALC >R >R >R DUP 1000 * R 100
 * + R> SWAP R 10 * + R> SWAP
 R + R> SWAP ;
 : LOW-LIMI 1000 OVER 100 SWAP
   */ ;

 0 VARIABLE ?DIGITS

 : DECODE DUP . OVER DUP . OVER
 U* D.
 DUP 22 < IF 4 ?DIGITS ! ELSE
 DUP 47 < IF 5 ?DIGITS ! ELSE
 6 ?DIGITS ! THEN THEN
 DUP DUP * U*  2DUP D.
 ?DIGITS @ 0 DO 10 M/MOD LOOP
 2DROP
 ?DIGITS @ 0 DO ZERO + C@ EMIT
 LOOP CR ;
 3 LOAD ;S

: TEST OVER OVER U* 10000 M/MOD
DROP 10 /MOD SWAP R-TRUE @ -
ABS 0 >
IF DROP DROP DROP
ELSE ZERO + DUP C@ 50 >
IF DROP DROP DROP
ELSE 100 SWAP C! 10 /MOD SWAP
ZERO + DUP C@ 50 >
IF DROP DROP DROP PURGE
ELSE 110 SWAP C! 10 /MOD SWAP
ZERO + DUP C@ 50 >
IF DROP DROP DROP PURGE
ELSE 111 SWAP C! 10 /MOD SWAP
ZERO + DUP C@ 50 >
IF DROP DROP DROP PURGE
ELSE 103 SWAP C! 10 /MOD DROP
ZERO + DUP C@ 50 >
IF DROP DROP PURGE
ELSE 97 SWAP C!
DECODE PURGE THEN THEN THEN
THEN THEN THEN ;
: R-SELECT 10 0 DO I I
ZERO + DUP C@ 50 >
IF DROP DROP ELSE 82 SWAP C!
DUP R-TRUE ! CALC LOW-LIMIT
100 SWAP DO I TEST LOOP DROP
ZERO + 45 SWAP C! THEN LOOP ;
                       4 LOAD ;S
: E-SELECT
10 0 DO I I ZERO + DUP C@ 50 >
IF DROP DROP ELSE 69 SWAP C!
R-SELECT
ZERO + 45 SWAP C! THEN LOOP ;

: S-SELECT
10 0 DO I I ZERO + DUP C@ 50 >
IF DROP DROP ELSE 83 SWAP C!
E-SELECT
ZERO + 45 SWAP C! THEN LOOP ;

: U-SELECT
10 2 DO I I
ZERO + 85 SWAP C!
S-SELECT
ZERO + 45 SWAP C! LOOP ;

: AUGUST-PUZZLE
ZERO 10 45 FILL CR
U-SELECT ;
```

```
Fill array with ASCII codes for minus sign (45)

DO "select & reserve U-digit between 2 & 9)"
    DO "select S-digit between 0 & 9"
        IF  "reject if occupied"
        ELSE "reserve it"
            DO "select E-digit between 0 & 9"
                IF  "reject if occupied"
                ELSE "reserve it"
                    DO "select R-digit between 0 & 9"
                        IF  "reject if occupied"
                        ELSE "charge R-TRUE"
                            "CALC user number"
                            "calculate LOW-LIMIT"
                            DO  "between LOW-LIMIT & 100"
                                "TEST--->DECODE"
                                "PURGE if needed"
                            LOOP
                            DROP "user number"
                        THEN
                    LOOP
                    "release E-digit"
                THEN
            LOOP
            "release S-digit"
        THEN
    LOOP
  "release U-digit"
LOOP
```

Structure of main program for August puzzle.

IF any reader does have serious delivery/non delivery problems with any supplier, whether or not they advertise in DU, we would like to know. Only rarely can we do anything to improve a genuinely dodgy situation (which fortunately are rare) but it helps us to build up a profile and identify any long-term problems. Come to think of it, that includes *Dragon User*, as, naturally, we want to know about any bottlenecks as early as possible. Apologies in advance for not acknowledging every letter, but where we *can* be of practical help, we will.

Winners and Losers

Every month, Gordon Lee will look at some prize programming points from a previous month's competition

WE wuz positively snowed under with replies to the March competition, many from readers entering for the first time — so who said it couldn't be done! Admittedly, the problem was a bit easier than many, as requested by a number of Dragon users who had thought that our usual competitions were a bit on the steep side. In fact, a number of readers found the problem a lot simpler than I had intended — an extra bonus! But more on that later.

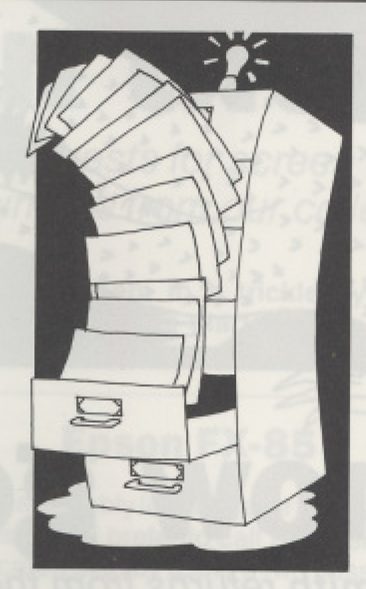
The only real difficulty that I can envisage was the 'ghost' space which is placed at the beginning of a string variable which is created by the computer when utilising the STR$ command. Thus X=12345 : x$=STR$(X) will produce a string variable, X$, which is six characters long, the first character of the string being a space in which there is room for an invisible + (positive) sign. In relation to the competition, if string variables are created in order to check the digits in certain positions, it has to be remembered that the first digit of the number is in the second position of the string. The easiest way of dealing with the problem is to remove the leading space immediately that the string has been created, by the use of the line X$=MID$(X$,2). This will then restore each digit in the string to its logical positioin. Only one entrant actually admitted to having been caught out by this at first after his original program failed to produce an answer.

Most readers realised that it was only necessary to test for numbers in the range 1000 to 3162 — that is those numbers whose squares have seven digits. However, many readers discovered simple proofs that the upper limit need be no greater that 1999, the final digit of the square must be '1', and that the third digit of lthe number is not greater than 3. I can do no better than quote from G.A.Hunt of Carnforth who writes:

"Let N be the four figure number and P be its square. Then since P is a seven figure number its square root must be less than 3142, and so the figure of N must be 1, 2, or 3. But only 1 is a possible last digit for a square. So the first figure of N must be 1. From this it follows that the first figure of P must be 1, 2, or 3, and hence the third figure of N must be 1,2, or 3. A perfect square ending in 1 can only be produced by being the square of a number ending in 1 or 9; hence the last digit of N must be a 1 or 9.'

This reduction in the number of possibilities requiring to be tested was a frequently expressed idea, with many entrants giving the running times of programs. The following is a representative sample:

Hugh Somerville: 3 hours
John Blatch: 2 hours
R. Crowther: 6.75 minutes
Simon Hargrave*: 49.5 seconds
James Bonfield*: 36 seconds
T.H. Denton: 10.96 seconds



(The * denotes the use of the 'speed poke' in the program).

The apparent simplicity of the competition certainly brought out the competitive spirit in many readers who, flushed with heady intoxication of so simple a task, came up with a variety of novel approaches! From R.Crowther of Rotherham came a solution in only one program line, while Chris Fry of West Wickham found a couple of alternative solutions were possible if the numbers were reversed in the grid. If the square is reversed the only solution is N=1000, while if N=2835 the grid can be filled if N is reversed and is entered as 5382.

Regular competitor Paul Weedon decided to eschew the conventional 'IF MID$(I$,3,1) = MID$(J$,1,1)' approach, and, with the reckless abandon of one who prefers to abseil despite the fact stairs are provided, came up with an alternative program utilising the powerful INSTR function. For readers who are not familiar with this command which searches for a 'target' string within a longer string, Paul has provided the following analysis of his routine:

I$ is the number and J$ is its square. Line 4 checks for the occurrence of the first digit of J$ starting from the third position in I$. If it is not there it goes to the NEXT I, otherwise it passes on the lines 5, 6, and 7, which check for the other three digits. Line * is a cross-check which if the other requirements are satisfactory, will give the required answer.

This goes to demonstrate that there is more than one way to swing a cat — or is it 'boil an egg'? Line 1 in this program is the 'speed' poke which should be omitted if your computer is not able to handle it. The INSTR command that Paul uses returns the position of the target string within the longer string. This can be demonstrated as follows:

X$="ABCDEFGHIJKLM";
Z=INSTR(X$,"F") : PRINT Z

This would give variable Z the value of 6 since the 'F' is at the 6th position in the string X$. A third parameter can be specified when using this function if you wish to search for the 'target' starting from a position inside the string to be searched. So, for example, Z=INSTR(7,X$,'F') would start the search at character number 7 in the string. in the above example, the target 'F' would not now be found and so the computer will ascribe a value of zero to variable Z.

Finally, I would like to thank the following for their good wishes: Mark Griffin of Dublin, Chris Fry of West Wickham, Graham Barber of Sutton Coldfield, Austan Henderson of Bromsgrove, and Lindsey Edwards, Chris Boutell, Mum, and Plod the cat, of Huntingdon.

R.Crowther's solution in one line:

```
1 FORN=1000TO3162:N$=STR$(N):S$=STR$(N*N):FOR
D=2TO8:N$(D)=MID$(N$,D,1):S$(D)=MID$(S$,D,1):
NEXT:IFN$(2)=S$(8)ANDN$(3)=S$(4)ANDN$(4)=S$(2
)ANDN$(5)=S$(6)THENPRINT" "N:PRINT"      "S$(3)
:PRINT"  "N:PRINT"      "S$(5):PRINTN:PRINT"
 "S$(7):PRINT"     "N ELSENEXT
```

Paul Weedon's solution using the INSTR function:

```
1 POKE&HFFD7,0
2 FOR I=1000 TO SQR(9999999)
3 I$=MID$(STR$(I),2):J$=MID$(STR$(I*I),2)
4 IF INSTR(3,I$,MID$(J$,1,1))<>3 THEN 9
5 IF INSTR(2,I$,MID$(J$,3,1))<>2 THEN 9
6 IF INSTR(4,I$,MID$(J$,5,1))<>4 THEN 9
7 IF INSTR(1,I$,MID$(J$,7,1))<>1 THEN 9
8 IF INSTR(1,J$,MID$(I$,3,1))+INSTR(3,J$,MID$(I$,2,1))+
  INSTR(5,J$,MID$(I$,4,1))+INSTR(7,J$,MID$(I$,1,1))=16
  THEN POKE&HFFD8,0:PRINT I:STOP
9 NEXT I
```



The Show goes on

*Late but lively, **Ken Smith** returns from the London 6809 Show*

THERE are times when you can be really pleased to be wrong. After all, wasn't I the one who said that the last 6809 Show probably would be just that. Yet only four months later there we were for yet another time. The Dragon and its devotees never cease to amaze me.

It is interesting to see the style of the show changing. This one seemed to me to have a more technical bias than previous shows, with probably more bargains than ever.

I closed the last report telling of how I wanted a modem, only to find the cupboard bare. Well true to form, this time my needs were supplied in the shape of a neat little Prism, supplied by the National Dragon User Group (I must join someday), complete with software for only forty pounds. Just proves my point, seek and ye shall find.

At the other end of the size scale Microdeal had a huge one, (modem that is) ex-GPO and bigger than the Dragon itself. Still at the silly prices they were asking you could always use it as a combined modem and bench seat. Maybe that is what is meant by sitting on information.

Bargains for grabs

Bargains come in all shapes and sizes and plenty were up for grabs on John Penn's stall.

One was of particular interest to prospective OS-9 users, namely the operating system itself which John is selling at cost price.

Anyone who has seen the ads from the gentleman in Denmark who has the licence to produce OS-9, will know the terrific prices he charges. Well, John's are a fraction of that and his only condition is that he would like you to buy you other software from him. With Pascal at only seventeen pounds, where else would we go?

Some new faces appeared on the stand taken by Broomsoft. A company that seem to deal mainly in adventure games, this was their first show and I hope to see more of them.

Unfortunately the larger houses and dealers seemed to get first bite at the cherry and a lot of people were spent out before they found Broomsoft, which is a pity, as they may never know what they missed. (*They might if they enter the DU June competition. Broomsoft have done us some discount vouchers as prizes. — Ed.*)

Keyboard basics

I would have been in a similar position had it not been for Bob Harris.

Bob markets the excellent BASIC42 with which I have been having a few minor problems which Bob thought could be down to my keyboard. This prompted me to go the the Peaksoft stand and try one of their replacement units. For those of you who, like myself, thought that the old Dragon 32 keyboard was good enough then you must try this.

The improvement in feel and response is appreciable and it seems to have cured my problem. The trouble is it doesn't spell any better than the old one. (*Too late! My last conversation with Peaksoft revealed that they have sold the last one and can't obtain any more stocks. A pity, as there is apparently a demand.*)

Computape were offering a wide variety of software and joysticks all at knockdown prices and in fact I can't remember a show with so many joysticks on offer.

Microdeal had a binful of Quickshots, to be sold on an "as seen" basis for one pound each. Some of their software was cheaper still. My youngest son managed to buy eight games for only four pounds.

Anyone with a little knowhow could have assembled a double drive disc system for a fraction of the normal price. NDUG had the drives, Compusense supplied the cases and DOS. They also had a lot of the disc based software, mainly FLEX orientated of course but some Dragon DOS and also some hardware, like an Eprom burner and their famous Plus board that puts you in the 128K league.

As I dodged from stand to stand I was surprised by the enthusiasm still offered by both the traders and the customers. This long after the collapse of Dragon Data you don't expect to see many new developments but there were. From conversions to give you a built in speaker and inverse screen at the flick of a switch, to the new game *Superkid* from Quickbeam.

One big surprise is that there are still some Dragoneers who don't subscribe to DU. (If you are reading a borrowed copy I hope you feel thoroughly ashamed of yourself. *Dragon User* is our strongest link and we must preserve it.)

The delightful Helen was there as promised, thrusting subscription forms at anyone who came within signing distance. Helen, by the way, did not disappear at twelve as threatened which explains the team of white mice and the soggy pumpkin I noticed on the way out. (*The silly chaps came to the wrong door, Ken. I had to flag down a passing banana drawn by a couple of cockroaches. — Ed.*)

Revitalised

I had arrived at 10.00, when the doors opened, with a sinking feeling (*Why? Were they poorly?— Ed. again*). I left just after 1.30, broke but revitalised. Please, oh please, let there be another show in November. I will need another fix of enthusiasm. Just to be in the company of people who really do believe in the Dragon might stop me going to one of those other, inferior, machines.

When you hear quotes like "after all it is still the only home micro capable of running the new education standard, namely OS-9," from someone who really believes in what he is saying, it makes all the effort worthwhile. See you in November.

■

Although it looks doubtful that there will be another show at the Horticultural Halls, we have heard from at least two lots of people who are thinking of putting on smaller shows in the autumn at easily accessible venues out of London — avoiding the swingeing city prices. The success of the Ossett Show in April, shortly after the 6809 Show, demonstrates that this format can work to everyone's advantage. We'll publish any news as soon as we get it.

Down in the dumps

***Dragon User** frequently gets requests for screen dumps. Here we present some specimens from our collection*

OK, you guys. You asked for it. The response to our appeal for screen dumps has come in a flood. We can only reproduce them in a trickle by comparison, but we're starting here. They aren't in any particular order, and they aren't guaranteed to work, but numbers of you should find what you are looking for eventually.

Epson FX-85

This is written in assembly language for the Epson FX-85. The routine handles what I prefer to call a half-size printout of a PMODE4 (3) screen display using a two pass technique, not unlike the NLQ-mode print.

Olav Havard Noraberg

```
2000                * HALF SIZE *
2000
2000                * GRAPHICS PMODE 3&4 SCREEN DUMP
2000                * ROUTINE  FOR  THE  EPSON FX-85
2000                * PRINTER
2000
2000  BCF5          TXLPCH EQU    $BCF5
2000  00BA          TOP    EQU    $BA
2000  00B7          BOTTOM EQU    $B7
2000
2000                * (PC)=POINTER TO LENGTH-BYTE
2000                * 0<NUMBER OF BYTES<255
2000
2000 3416           LPINIT PSHS   A,B,X
2002 AE64                  LDX    $4,S
2004 E680                  LDB    ,X+
2006 2708           LOOP1  BEQ    END
2008 A680                  LDA    ,X+
200A BDBCF5                JSR    TXLPCH
200D 5A                    DECB
200E 20F6                  BRA    LOOP1
2010 AF64           END    STX    $4,S
2012 3596                  PULS   A,B,X,PC
2014
2014                * STORES 8 BYTES IN THE USER
2014                * STACK AREA (X=BASE, Y=Y+64)
2014
2014 3444           PUT    PSHS   B,U
2016 1F12                  TFR    X,Y
2018 C608                  LDB    #$08
201A 63A4           LOOP2  COM    ,Y
201C A6A4                  LDA    ,Y
201E 3602                  PSHU   A
2020 31A840                LEAY   $40,Y
2023 5A                    DECB
2024 26F4                  BNE    LOOP2
2026 35C4                  PULS   B,U,PC
2028
2028                * ROLL 8 BITS INTO A-REG FROM
2028                * 8 BYTES IN THE USER STACK AREA
2028
2028 3444           ROLLBY PSHS   B,U
202A C608                  LDB    #$08
202C 68C2           LOOP3  LSL    ,-U
202E 49                    ROLA
202F 5A                    DECB
2030 26FA                  BNE    LOOP3
2032 35C4                  PULS   B,U,PC
2034
2034
2034 3477           @MAIN  PSHS   A,B,CC,X,Y,U
2036 9EBA                  LDX    TOP
2038                * RESERVE 8 USER STACK BYTES
2038 1F43                  TFR    S,U
203A 3278                  LEAS   -8,S
203C                *-------------------------------1
203C 8DC2           LOOP4  BSR    LPINIT
203E                * LINE SPACING = 23/216", CR,
203E                * PLOTTER GRAPHICS = 256 BYTES
203E 091B33170D            FCB    $09,$1B,$33,$17,$0D
2043 1B2A070001            FCB    $1B,$2A,$07,$00,$01
2048 C640                  LDB    #$40
204A                *-------------------------------2
204A 3404           LOOP5  PSHS   B
204C 8DC6                  BSR    PUT
204E                *-------------------------------3
204E C608                  LDB    #$08
2050 8DD6           LOOP6  BSR    ROLLBY
2052 BDBCF5                JSR    TXLPCH
2055 5A                    DECB
2056 26F8                  BNE    LOOP6
2058                *-------------------------------3
2058 3001                  LEAX   1,X
205A 3504                  PULS   B
205C 5A                    DECB
205D C120                  CMPB   #$20
205F 260C                  BNE    CONT
2061 8D9D                  BSR    LPINIT
2063                * LINE SPACING = 1/216", CR,
2063                * PLOTTER GRAPHICS = 256 BYTES
2063 091B33010D            FCB    $09,$1B,$33,$01,$0D
2068 1B2A070001            FCB    $1B,$2A,$07,$00,$01
206D 5D             CONT   TSTB
206E 26DA                  BNE    LOOP5
2070                *-------------------------------2
2070 308901C0              LEAX   $01C0,X
2074 9CB7                  CMPX   BOTTOM
2076 25C4                  BLO    LOOP4
2078                *-------------------------------1
2078                * CLEAN UP STACK
2078 3268                  LEAS   $8,S
207A 8D84                  BSR    LPINIT
207C                * SET LINE SPACING = 1/6"
207C 021B32                FCB    $02,$1B,$32
207F                * INVERT SCREEN DISPLAY
207F 9EBA                  LDX    TOP
2081 6380           LOOP7  COM    ,X+
2083 9CB7                  CMPX   BOTTOM
2085 25FA                  BLO    LOOP7
2087 35F7                  PULS   A,B,CC,X,Y,U,PC
```

OKI-80 Microline

This was very complicated to make, because you have to turn the screen to have it fit the paper. Secondly, the screen bytes and the printer bytes are configured differently.

Jakob Hoffmann

```
4E21 9EBA              LDX   $BA
4E23 30891801          LEAX  6145,X
4E27 6382      INVERS  COM   ,-X
4E29 9CBA              CMPX  $BA
4E2B 26FA              BNE   INVERS
4E2D 86BF      DUMP    LDA   #191
4E2F B74EF1            STA   YPOS
4E32 7F4EF0            CLR   XPOS
4E35 108E4EF3          LDY   #SKAERM
4E39 8D0A      MAIN    BSR   GET
4E3B 108C6EF3          CMPY  #SKAERM+8192
4E3F 26F8              BNE   MAIN
4E41 170077            LBSR  PRINT
4E44 39                RTS
4E45 8D47      GET     BSR   PPOINT    ;X,Y
4E47 7A4EF1            DEC   YPOS      ;X,Y-1
4E4A 8D42              BSR   PPOINT
4E4C 7C4EF1            INC   YPOS      ;X+1,Y
4E4F 7C4EF0            INC   XPOS
4E52 8D3A              BSR   PPOINT
4E54 7A4EF1            DEC   YPOS      ;X+1,Y-1
4E57 8D35              BSR   PPOINT
4E59 7C4EF0            INC   XPOS      ;X+2,Y
4E5C 7C4EF1            INC   YPOS
4E5F 8D2D              BSR   PPOINT
4E61 7A4EF1            DEC   YPOS      ;X+2,Y-1
4E64 8D28              BSR   PPOINT
4E66 B64EF2            LDA   BUFF
4E69 46                RORA
4E6A 1A01              ORCC  #1
4E6C 46                RORA
4E6D A7A0              STA   ,Y+
4E6F 7A4EF0            DEC   XPOS
4E72 7A4EF0            DEC   XPOS
4E75 7A4EF1    LOOP    DEC   YPOS
4E78 B64EF1            LDA   YPOS
4E7B 4C                INCA
4E7C 2701              BEQ   MERE
4E7E 39                RTS
4E7F 86BF      MERE    LDA   #191
4E81 B74EF1            STA   YPOS
4E84 7C4EF0            INC   XPOS
4E87 7C4EF0            INC   XPOS
4E8A 7C4EF0            INC   XPOS
4E8D 39                RTS
4E8E 8620      PPOINT  LDA   #32
4E90 F64EF1            LDB   YPOS
4E93 3D                MUL
4E94 D3BA              ADDD  $BA
4E96 1F01              TFR   D,X
4E98 F64EF0            LDB   XPOS
4E9B 54                LSRB
4E9C 54                LSRB
4E9D 54                LSRB
4E9E 3A                ABX
4E9F BF4EEE            STX   ADDR
4EA2 B64EF2            LDA   BUFF
4EA5 F64EF0    RUL     LDB   XPOS
4EA8 C407              ANDB  #7
4EAA 5C                INCB
4EAB F74EF2            STB   BUFF
4EAE E684              LDB   ,X
4EB0 59        AGAIN   ROLB
4EB1 7A4EF2            DEC   BUFF
4EB4 26FA              BNE   AGAIN
4EB6 46                RORA
4EB7 B74EF2            STA   BUFF
4EBA 39                RTS
4EBB 861B      PRINT   LDA   #27
4EBD BD800F            JSR   $800F
4EC0 8642              LDA   #66
4EC2 BD800F            JSR   $800F
4EC5 861D              LDA   #29
4EC7 BD800F            JSR   $800F
4ECA 861B              LDA   #27
4ECC BD800F            JSR   $800F
4ECF 8638              LDA   #56
4ED1 BD800F            JSR   $800F
4ED4 8660              LDA   #96
4ED6 979B              STA   $9B
4ED8 7F0148            CLR   $148
4EDB 860D              LDA   #13
4EDD BD800F            JSR   $800F
4EE0 8E4EF3            LDX   #SKAERM
4EE3 A680      ONLINE  LDA   ,X+
4EE5 BD800F            JSR   $800F
4EE8 8C6EF3            CMPX  #SKAERM+8192
4EEB 26F6              BNE   ONLINE
4EED 39                RTS
4EEE 0000      ADDR    FDB   0
4EF0 00        XPOS    FCB   0
4EF1 00        YPOS    FCB   0
4EF2 00        BUFF    FCB   0
4EF3           SKAERM  RMB   8192
6EF3
```

Tandy CGP-115 plotter

This one is speaking for itself...

John Oliver

AT LAST HERE IS A SCREEN DUMP PROGRAM FOR ALL DRAGON OWNERS WHO POSSESS A TANDY CGP-115 4 PEN PRINTER PLOTTER.
THIS IS ALSO A LATE RESPONSE IN REPLY TO REQUESTS IN THE communication PAGE OF D.USER EDITIONS JUNE & OCTOBER 1986, BY C.CAREY OF EASTBOURNE GROVE,ESSEX AND D.S PHILIPS OF BROADSTONE ROAD SOUTH, CHESHIRE.

THE PROGRAM IS SET UP FOR PMODE 4 WITH EITHER SCREEN ALSO ASSUMING THE 1ST 4 GRAPHICS PAGES ARE USED.
HERE ARE A FEW PROGRAM NOTES.
LINE 10:-SETS PRINTER TO GRAPHIC MODE WITH THE ORIGIN SET AT 480,0
LINE 20:-DISPLAYS SCREEN.

LINES 30-50:-INVERT THE SCREEN ONLY IF THERE IS MORE BLACK THAN WHITE SO SAVING ON YOUR BLACK INK.
LINE 60:-LOOP EQUAL TO THE NUMBER OF VERTICAL DOTS ON SCREEN. THIS MAY BE ALTERED FOR OTHER RESOLUTIONS.
LINE 70:-AS 60 BUT FOR THE NUMBER OF HORIZONTAL DOTS.
LINE80:-PUTS COLOUR OF DOT LOOKED AT BY THE TWO LOOPS INTO VARIABLE (P).
LINE90:-PRINTS OUT A 2 DOT LENGTH LINE FOR EACH BLACK DOT ON THE SCREEN
NOTE THE COLOUR CODE MAY BE ALTERED FOR OTHER MODES,USUALLY BEST TO CHANGE TO THE MAIN COLOUR USED,AND DO NOT FORGET TO CHANGE YOUR PRINTERS PEN COLOUR!
LINE 100:-END OF HORIZONTAL DOT LOOP.
LINE 110:-MOVES PAPER BACK READY FOR NEXT LINE TO BE PRINTED ALSO MOVES PEN.
LINE 120:-END OF VERTICAL DOT LOOP.
LINE 130:-FEEDS FINISHED DUMP TO YOU.

If you want a darker screendump then enter the following directly without a linenumber or remark.
For this to work correctly YOU SHOULD NOT have to move the paper by hand/feed. YOU CAN REPEAT THIS AS MANY TIMES AS YOU WISH TO OBTAIN YOUR REQUIRED DUMP.

```
1 REM:D=0:?#-2,"H":?#-2,"M-1,0":?#-2,"I"
:GOTO 60
```

FURTHER NOTE THAT IN PMODES 1&3 LINES 30-50 MAY BE EITHER REMOVED OR TURNED INTO REMARKS.
IN PMODE 0 LINES 30&50 SHOULD BE:-
FOR K=1536 TO 3071: ETC. LINE 40 BECOMES
IF Q>=195840 THEN 60
IN PMODE 2 CHANGE 30&50 TO BECOME
FOR K=1536 TO 4607:ETC. LINE 40 BECOMES
IF Q>=391680 THEN 60

```
10 PRINT#-2,CHR$(18):PRINT#-2,"M480,0":P
RINT#-2,"I"
20 PMODE4,1:SCREEN1,1
30 FOR K=1536 TO 7680:Q=Q+PEEK(K):NEXT K
40 IF Q>=783360 THEN 60
50 FOR K=1536 TO 7680:POKE K,255-PEEK(K)
:NEXT K
60 FOR Y=0 TO 191
70 FOR X=0 TO 255
80 P=PPOINT(X,Y)
90 IF P=0 THEN PRINT#-2,"J0,-2" ELSE PRI
NT#-2,"R0,-2"
100 NEXT X
110 D=D-2:PRINT#-2,"M"+STR$(D)+",0"
120 NEXT Y
130 PRINT#-2,"M0,-600":END
```

Dragon Answers

If you've got a technical question write to Brian Cadge. Please do not send a SAE as Brian cannot guarantee to answer individual inquiries.

A sound start

I AM involved in a sports club and we want to use my Dragon to time races etc. I have written a Basic program to do this but I would also like the computer to start the races with a tone. Can you explain if and how I can send sound to the cassette port and how to connect an amplifier to it. Also will I be able to send separate sounds to the TV and PA systems?

Dick Tabber
Reading

THE output from the cassette port is suitable for directly driving a standard amp (not a phono input!). Looking at the Dragon's cassette socket end on and numbering the pins anticlockwise 1 to 5, you need to connect pin 2 to the input line of your amp and pin 3 to the ground line.

All sound produced by the SOUND and PLAY commands is automatically sent to this port so you need do no more. It is possible to select between TV/Cassette Port and Cassette Port only, however, this cannot be done from Basic as the SOUND/PLAY commands always enable TV sound.

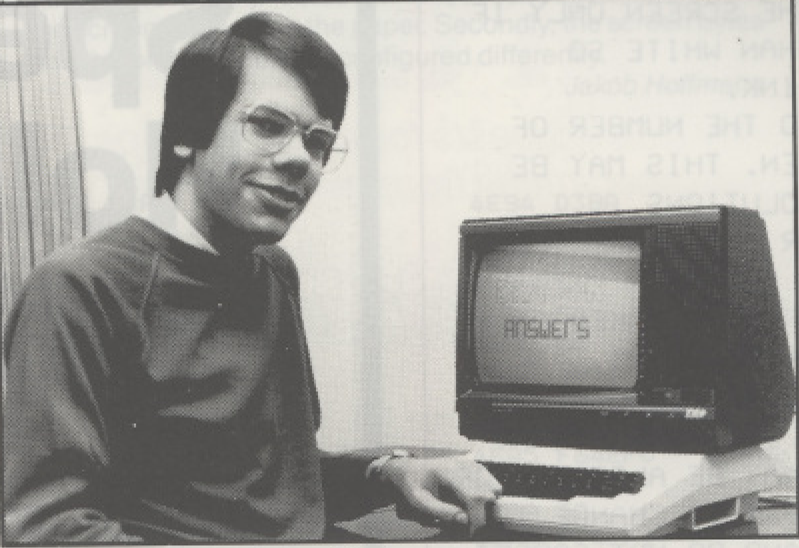
Flex or OS-9?

I AM a student presently studying a computer course. Having owned a Dragon for four years and using it mainly for assembley language programming I find the machine excellent for my needs but I would like to obtain a 'commercial' operating system for it. The alternatives would apper to be Flex or OS-9. Flex, although preferable, is the more expensive. I therefore would like to obtain OS-9 but this seems unobtainable?

C.P. Goodwin
5 North Devon Rd.
Fishponds
Bristol

THERE has been much (heated) debate over the years in *Dragon User* about the relative merits of OS-9 and Flex. Both are now available once again for the Dragon 64: Flex from Compusense is, in fact, the cheaper of the two at £99.99 which includes an excellent editor/assembler package plus a disc version of Basic. OS-9 from H.C. Anderson also comes with an excellent editor/assembler and debugger (but no Basic) and is somewhat more expensive at £174.

It must be up to the individual to weigh up the relative aspects of these two very different operating systems, but Flex is cheaper and does have a larger software base.



Colour runs out

MY Dragon 32 loses its colour transmission after 5 to 10 minutes running time. It functions correctly in other respects, can you please advise me?

John Derry, Leicester

THE most common cause of this problem is that, along with many other home computers, the Dragon tends to 'wander' slightly off channel during operation. Some TVs are more sensitive to this than others (some of the more modern ones will keep re-tuning themselves to the strongest signal).

If you find that re-tuning does not restore the colour then the chances are that the Dragon itself, specifically the modulator, is at fault. This is relatively simple to replace yourself (with a soldering iron and a little skill) or alternatively your local TV engineer will probably do it for you.

Cure for Anadex

I RECENTLY acquired an Anadex printer and I was wondering if it is possible to connect it to my Dragon 32. It is a Model DP-8000 dot matrix. There are two connections at the rear of the printer and 3 sets of dip switches inside. There are also four switches and a light on the front; on/off line, linefeed, TOF etc. I have no knowledge whatsoever of printers.

Richard Ball
75 Sunnybank Rd.
Pontypool, Gwent

THE two sockets at the rear of the printer are for serial and parallel interfaces. The lower socket (a 36 pin female type) is a standard centronics interface as used on the Dragon. Any standard Dragon printer lead should work with this printer.

In the first bank of dip switches, 1-5 control the form length in increments of 0.5 inch; 6 controls regular/condensed font; 7 controls the print time-out function; 8 controls the selection of '£' or '£' in the character set.

In the second bank of dip switches, 1-2 control the skip over perforation length in increments of 0.5 inch; 4 sets 6 or 8 lines per inch; 7 controls wrap-round/truncate mode and should be OFF for the Dragon; 8 controls the auto linefeed setting and should be ON for the Dragon.

The third bank of 8 dip switches is concerned with setting the baud rate of the serial port and should all be 'off' when using the parallel port.

Stood him up

I AM writing a basic program to run from disc. I want the program to display the date at which it was last run, but I cannot see any way of keeping the date (inDT$) with the program as when I save it to disc and then reload, the variables appear to be cleared.

Paul Dreron
Nottingham

ONE solution to this would be to save the date in a disc file and reload it at the start of the program.

Another, more interesting solution, is to store the date permanently in your program which can then be saved to disc before exiting. Assuming the variable DT$ holds the date then the following code could be used:

```
100 X$="THIS PROGRAM WAS
LAST RUN ON 00/00/00"
110 PRINT X$
120 INPUT "ENTER TODAY'S
DATE DD/MM/YY";DT$
130 X=VARPTR(X$)
140 A=PEEK(X+2)*256+
PEEK(X+3)
150 FOR I=1 TO 8:POKE A+I+28,
ASC(MID$(DT$,I,1)):NEXT I
160 SAVE "PROGNAME"
170 REM REST OF PROGRAM....
```

This works by poking the new date directly into the string X$ (whose address is found by using the Varptr function). When a string is declared as a constant (as in line 100), basic does not waste string space on it but sets its pointer to point directly in the program text, hence it is permanently changed and saved along with the rest of the program in line 160.



Super SAM

Matthew Lodge is here to tell you about a double-dealing chip

THE Synchronous Address Multiplexer (chip number 6883 or 74LS783), to give it its full title, has never been fully explained in one article. However, it is one of the most important chips in the Dragon, and unfortunately one of the most prone to breakage. This breakage can be accelerated by bad software practice, so beware. The infamous speed-up poke WILL DAMAGE YOUR COMPUTER. The local Dragon know-all will tell you that it doesn't matter, and some software even uses it to achieve faster running speeds but IT WILL KILL YOUR DRAGON. The Dragon was not designed to run at the faster speed and it will cost you dearly in time and money if you use the poke (Compusense supply the 6883/74LS783 for £25 a time).

Basically, SAM provides most of the address decoding for the 6809 chip, or in very simple terms, what goes where. SAM starts at $FFCO and finishes at $FFDF. Some of the facilities are useful on their own, others are used in conjunction with the two PIA chips on the Dragon. First I'll explain how to access SAM.

Each register in SAM has two locations allocated to it. One even, and one odd. The odd location turns on the register, the even one turns it off. This dual location system also has the side effect that you can't tell how SAM is set up by peeking to it, because all you get back is $7E (126).

For example, let's take the memory map toggle, locations $FFDF and $FFDE. The location $FFDE is even (if you can't see why, convert it to decimal). $FFDF is odd. When you type

POKE &HFFDF,0

you are setting the map type to one. The number you poke doesn't matter at all. When you type

POKE &HFFDE,0

you reset it to zero. But don't do it yet! Too late, you did it? That's OK for all with a Dragon 32, but if you have a Dragon 64 then you've got a crash. This is because on the Dragon 64, map type one is RAM mode, that's 64K of continuous RAM, and map type zero is the normal 32K mode. And because you were running Basic, all the interrupts were pointing to areas in the Dragon ROM. When you changed the map type, the ROM disappeared, and the interrupts had nowhere to go. Also, you were running Basic, which is a program just like any other, and the program itself also disappeared, so wallop!, you have a crash. On the Dragon 32, map type one does not apply, so nothing happened. (Unless you have a very rare and early Dragon 32 which has 64K of RAM fitted, but only 32K used).

This register is useful, as '64 owners can utilise it from assembly language programs. In order to prevent a crash, all you have to do is turn off the interrupts (ORCC £$50), then change modes. If you want to change back, all you have to do is change modes back, then re-enable them (ANDCC £$AF). Thus, it is possible to copy 32K Basic and DOS into RAM where they can be hacked, sorry, altered (a program to do this is listed as program one). This is the method that Harris's Basic 42 uses to hook in all of its goodies.

The '64 owners among you may now be wondering "but what happens when I type EXEC 48000 to go into 64K mode?". What happens is that a small machine code program is copied into the cassette buffer, in low RAM. This turns off the interrupts, then switches in the second ROM (which appears at $8000 to $BFFF) and copies some code from it. Then it switches in map one, which is 64K RAM mode. This still leaves the cassette buffer and all low RAM (The first 32K) intact. It stores this code in its new position (ie +$4000 from where it got it), then repeats the operation until all of the Basic is transferred, then it 'boots' the new Basic (that is not a machine code kick, but a term meaning roughly the same as RUN).

Look at **figure one**. This provides a sketch of what each toggle means to SAM. I'll describe what each means, from $FFC0 up.

$FFC0 to $FFC5 is the video graphics mode. This is used in conjunction with PIA, and I am not going to explain here how to access all the modes, as it takes a long time. If you really want to know then contact Motorola for a SAM/68477 applications guide (free) which gives all the necessary diagrams, or get a copy of Roy Coates's article on the Graphic modes in November '85 *Dragon User*. Having said that, I'm going to try to explain it in simple terms here.

These locations control the VDG memory modes, or if you like, bytes per page.

MODE BINARY REAL TERMS
0 000 512
1 001 1024
2 010 2048
3 011 1536
4 100 3072
5 101 3072
6 110 6144
7 111 7680 Watch out Picasso!

Don't mix pages in this sense with pages in the Dragon Basic sense. A page here means the number of bytes used by the screen. A page in Dragon Basic means 1536 bytes. So, 6 or 6144 bytes is correct for PMODEs 3 and 4. Using the binary column from the table above (working from left to right or down in memory), and the SAM diagram, poke the odd location to set one bit, and poke the even location to clear a bit. So if you wanted to set 1536 bytes per page, you would:

POKE &HFFC4,0:POKE &HFFC3,0:
POKE &HFFC1,0

— which are the actual registers. 0,1,1 are the binary equivalents.

However, don't think that poking to $FFC0, $FFC3 and $FFC5 will get you pmode 4. You haven't set the PIA up, or the display offset. The latter just happens to be next in memory.

The locations from $FFC6 to $FFS3 control the start of the VDG mode you are currently in, that is the address at the top left hand corner of the screen. While in Basic it's 1024, but it can be anything from 0 to $FE00. You notice that there are only seven bits, so only addresses in multiples of $200 (512) apply. There goes your fast hardware scroll (a technique often used on the BBC, which uses the 6847's sister chip, the 68450). And before you rush off and write an animation program which flips through frames by moving the display offset, note that some Dragons have a hardware error, so that pages over $4000 cannot be assigned. All you get if you try is lovely green swaying garbage. The machine and all the RAM is OK, it's just that your display is...different. I've just killed two potentially good ideas stone dead, but moving the

Program 1:

```
START   ORCC    £$50     INTERRUPTS OFF
        LDX     £$8000   START OF BASIC
LOOP    STA     $FFDE    32K MODE
        LDD     ,X       LOAD NEXT WORD FROM ROM
        STA     $FFDF    64K MODE
        STD     ,X++     STORE IN SAME PLACE,
                         BUT IN RAM
        CMPX    £$FF00   END OF DOS ?
        BLO     LOOP     NOPE, SO DO NEXT WORD
        ANDCC   £$AF     INTERRUPTS ON
        RTS              FINISH (NOTE: EXITS IN
                         64K RAM MODE)
```

display does have its uses. If you have a machine code graphics program which needs all the space it can get, try locating the screen at 1024, or even put it at 512. If it doesn't use any ROM calls or only a few simple ones, then why not locate it at 0. If you are using text mode, you can fit quite a few separate text screens in $4000 bytes.

Now we come to locations $FFD4 and $FFD5, which is labelled page on the diagram, but it doesn't do anything when in 64K mode. I must confess I don't know anything of value about this location. All it has on my proto diagram is "MPU addresses from $0000 to $7FFF apply to page £1 if P1 = "1"".

Next we come to the legendary MPU rate. This is the thing all the bad software writers twiddle when they want to turbocharge their Dragons. The Dragon operates in slow mode normally, but three other modes exist. The first is just fast, the second is superfast and the third is turbocharged. You will notice that you lose the display if you try superfast or turbo. This is because the machine is running so quickly it doesn't give the poor VDG any processor time to do its work, so it is left bewildered, with the swaying green garbage. Underneath all that, the machine is still working although the keyboard won't be, as it was designed to work at a slower speed.

Next is $FFDA to $FFDD which is memory size. There are four possible values (2 bits), all of which are marked in the diagram. The Dragon uses the third mode all the time. If you try telling the computer it has 64K of static RAM, interesting things result, with a crash which kills the systems.

One last comment, which is that all registers in SAM are cleared when the machine is reset. That concludes SAM, and I hope this has provided a helping hand to any programmer bewildered by the SAM chip. I should like to express my thanks to Compusense, without whom, etc., etc.

Glossary

Interrupts: Every so often the 6809 is interrupted. That is, it finishes executing the current instruction, then services a small machine code routine. When it reaches the RTI instruction, it returns to whatever it was doing before. On the Dragon interrupts happen every 50th of a second, and are caused by the VDG chip which is signalling that it is starting to scan the TV screen. The Basic ROM uses this to update the software clock or timer so it can keep track of real time (useful when PLAYING notes, to see when to stop).

MPU: Micro Processing Unit

SAM: Synchronous Address Multiplexer

Static RAM: Normal RAM (as used on the Dragon) needs what is called a refresh signal every few milliseconds. SAM provides this, and it serves to refresh the contents of the RAM or make it remember what it holds. Static RAM does not need a refresh signal. The price you pay is in cost and size, as static RAM chips cost more and there is less memory to a chip (the extra circuitry on the square of silicon takes up quite alot of room).

VDG: Video Generator

6847: VDG used in the Dragon. Designed to be used in conjunction with SAM, which explains why some circuitry it needs is on SAM.

6845 CRT controller (different to a VDG) used in the BBC micro and the IBM PC (Spit!)

6883: SAM's chip number

74LS783: SAM's more common chip number (one of a series of 74LSxxx chips).

Figure 1 – SAM proto diagram

C=CLEAR BIT
S=SET BIT

All bits cleared when SAM reset

Address	C/S	Bit	Function
FFC0	C	V0	VDG
FFC1	S	V0	VDG
FFC2	C	V1	MODE
FFC3	S	V1	MODE
FFC4	C	V2	(SAM)
FFC5	S	V2	(SAM)
FFC6	C	F0	DISPLAY OFFSET (BINARY) Address of uppper left-most display element
FFC7	S	F0	
FFC8	C	F1	
FFC9	S	F1	
FFCA	C	F2	
FFCB	S	F2	
FFCC	C	F3	
FFCD	S	F3	
FFCE	C	F4	
FFCF	S	F4	
FFD0	C	F5	
FFD1	S	F5	
FFD2	C	F6	
FFD3	S	F6	
FFD4	C	P1	PAGE #1 — MPU ADDRESSES FROM $0000 TO $7FFF APPLY TO PAGE #1
FFD5	S	P1	
FFD6	C	R0	MPU
FFD7	S	R0	MPU
FFD8	C	R1	RATE
FFD9	S	R1	RATE
FFDA	C	M0	MEMORY
FFDB	S	M0	MEMORY
FFDC	C	M1	SIZE
FFDD	S	M1	SIZE
FFDE	C	TY	MAP TYPE
FFDF	S	TY	MAP TYPE

VDG modes:

Bit	AI.AE.S4.S6	G1C.G1R	G2C	G2R	G3C	G3R	G6R.G6C	DMA
V0	0	1	0	1	0	1	0	1
V1	0	0	1	1	0	0	1	1
V2	0	0	0	0	1	1	1	1

MPU rate:

Bit	-FAST	-FAST	-A.D.	-SLOW
R0	1	0	1	0
R1	1	1	0	0

Memory size:

Bit	64KS	64KD	16K	4K
M0	1.	0	1	0
M1	1	1	0	0

Electronic Author

Philip Beed reviews Quickbeam's popular wordprocessor

DESCRIBED in its adverts as "The Dragon Word Processor" *Electronic Author* certainly has a lot to commend it. It combines ease of use with some very powerful features. However as with all good things there is a drawback. The manual is not as comprehensive as it could be, it lacks features which could easily be included, and unfortunately it has some bugs. The objective of this article is to provide users with more information on using this word processor. If it also persuades Wayne Smithson to take it back to the drawing board and bring out an enhanced version then that can only be a good thing. Most of my work has been with the disc version of EA and while some of this article will relate specifically to discs, other parts will be applicable to tape and disc versions.

ENTER As stated, pressing ENTER is not actually necessary and slows EA down. However, it can be very useful for such things as adding underlines and boxing titles (see **figure 1**). To ensure your lines are the same length, press ENTER so that you type the lines directly below one another. When everything is as you want it cancel the ENTERS.

TAB This command is again skipped over very lightly in the manual, what it in fact does is tab on n spaces (in literal mode only), to use it to reach a specific column (on a new line). Concurrent use of the £T command is a far better way of making an address block than the method demonstrated in the manual.

MARGINS and **LINE LENGTH** Although you must sick to your original top and bottom space, you are at liberty to alter other settings, namely margin and line length. Uses of this would be when changing print font to a font with more or less characters per inch, thereby retaining the same size margins (see **figure 2**), and for indenting a body of text (see **figure 3**). A point to make clear is when using double width printing although the margin is taken care of line length is not so care needs to be taken to avoid wordwrap and untidy results. The best method is to ensure you do not exceed half the normal line length and issue a new line command immediately after cancelling double width.

PAGE NUMBERING It might have been better if off page numbers had been made the default, it is important to remember to put an £OP at the top of any text if you do not want page numbers. Page numbers do not reset if you load new text so if you start work on a second document using page numbers you may be surprised to find the numbers do not start at one, the answer is to assume nothing and always use £PN1. To avoid unwittingly missing out any parameter settings I have saved several of my most commonly used settings and load them adding the text to them either from the keyboard or by appending a file. The most common reasons for not getting the desired printout are, forgetting to put a cancelling command or a £ at the end of a centered line, and failing to leave gaps after format commands. These are the things to check for when checking your text.

LOADING FILES An undocumented property of the cassette file loading command is it will load virtually any cassette file not just those written by EA, all files saved by basic file writing programs can be loaded eg basic wordprocessors and databases, and some machine code files for example those written by DRS. The benefits of this could be enormous, users upgrading from a basic wordprocessor can retain there files and transfer them to disc, database information can be loaded, edited and saved in a text file format.

Figures 1, 2 & 3:

```
FIG 1.

                 ######################
                 #   BOXED  TITLES    #
                 ######################

FIG 2.

THIS TEXT IS IN NORMAL MODE AND IS AT 10 CHARACTERS PER INCH
WITH A MARGIN OF TEN CHARCTERS LEFT AND WRITE.
WE ARE NOW IN CONDENSED MODE WITH 17 CHARACTERS PER INCH BUT AS YOU ACN SEE THE MARGINS REMAIN THE
SAME AS I ALTERED THE FORMAT COMMANDS TO ACCOUNT FOR THE MORE COMPACT PRINTING.
AND NOW WE RETURN TO NORMAL MODE, IT IS VERY IMPORTANT TO DO
THIS IF YOU DO NOT WANT YOUR TEXT TO BE MISALIGNED WHEN
USING DIFFERENT FONTS.


FIG 3.

THE NEXT DEMONDRATION IS OF INDENTED TEXT, A GOOD EXAMPLE OF
THIS WOULD BE TO INCORPORATE A QUOTE IN YOUR BODY OF TEXT:-

      GOOD WORD PROCESSORS SHOULD INCORPRATE
      EASE OF USE WITH FLEXIBILITY AND FULL
      CONTROL OVER EVENTUAL TEXT LAYOUT ON
      PAPER!

NOW WE RETURN TO THE MAIN BODY OF THE TEXT AGAIN, THIS CAN
BE VERY USEFUL FOR MAKING PARTS OF THE TEXT STAND OUT.
```

LITERAL PRINTING The problem with justified text is it does not allow you to line up items of text. The answer to this is to switch off justification for text where this is necessary, the manual does not make it clear you can do this. For example to produce a list of items as in **figure 4** use £LI type the items, remembering you will have to take care to prevent word wrap yourself so do not exceed the line length, and add a £B at the end of each line. Remember also my tip for using ENTER to line things up. On completion use £FJ to return to justified mode. £LI does not affect any of the parameters you have set, so margins, page lenght etc. will remain correct.

BUGS Many of these are quite serious and not what one would expect from a program costing £ 20.

REPLACE FILE For some reason the replace file does not leave a backup file, this is not a problem in itself but for some reason the replace file option is sometimes corrupt. I say sometimes as it often works correctly, but on many occasions it reports "FILE EXISTS" then gives DOS ERROR 158 then reverts back to the text file screen, but while doing so kills the file on disc. There is no indication to the user that this has been done except a slight whirr of the drive head. Anyone not paying 100 % attention could easlily be fooled into thinking the file had been written to disc, NEW the file, and then save a subsequent file to the disc, losing the new version and the old version, which could well represent many hours work.

WRITE PROTECT If you try to write your file to a protected disc, an easy mistake especially if you have taken a base file from a protected disc, the error trapping reports a problem but then exits EA losing everything for you. There is in fact a way out of this (and other crashes or if you press reset). EXEC 10259 renters program, then press BREAK and use the old command to recover your text, I would then suggest saving the text immediately and reloading EA as it may well have corrupted. This may not work for all crashes and should not even be necessary — a little bit of error trapping would have seen to that.

FILENAME C EA will allow you to save filenames beginning with C but then refuses to reload them as it thinks you want to load a cassette file. The answer is to exit EA rename the file, a better answer would be for EA not to allow filenames beginning C.

PROGRAM DEFICIENCIES EA lacks in the following ways: It does not allow disc users to save files to cassette, uses for this would be backup, and sending to other users with the tape version. Directory scrolls off the top of the screen without giving the user a chance to see the top files, and with no pause facility. Disc output is restricted to single drive use which is extremely limiting for users with a dual drive as file manipulation can take a unnecessary amount of time in disk swapping while drive two remains idol. Lack of space cannot be used as an excuse as a separate disc control routine could easily be hooked in by the main program. No option for exiting to Basic is provided, this could be very useful for such things as renaming files or loading other utilities in conjunction with EA (eg address book) without having to switch off the Dragon.

```
# FIG 4.TO PRODUCE A NEATLY ALIGNED LIST OF ITEMS
WE NEED TO ENTER LITERAL MODE :-

1. JETSET WILLY      - PLATFORM,           FIVE DRAGONS
2. SPEED RACER       - GRAND PRIX,         FIVE DRAGONS
3. WORLDS OF FLIGHT - FLIGHT SIMULATOR, FOUR DARGONS

NOW SEE WHAT HAPPENS IN JUSTIFY MODE:-

1. JETSET WILLY - PLATFORM, FIVE DRAGONS
2. SPEED RACER - GRAND PRIX, FIVE DRAGONS
3. WORLDS OF FLIGHT - FLIGHT SIMULATOR, FOUR DARGONS
```

```
FIG 4. Literal printout of the file used to
create examples:-

 #TS4#BS4#PL66#LM10#LL60#OP#FJ FIG 1.#B1#DW#CT@@CE3
 ##################@# BOXED TITLES #@################@
 @CT##CD#B2 FIG 2.#B1 THIS TEXT IS IN NORMAL MODE AN
 D IS AT 10 CHARACTERS PER INCH WITH A MARGIN OF TEN
  CHARCTERS LEFT AND WRITE.#B#C+#LM17#LL99 WE ARE NO
 W IN CONDENSED MODE WITH 17 CHARACTERS PER INCH BUT
  AS YOU ACN SEE THE MARGINS REMAIN THE SAME AS I AL
 TERED THE FORMAT COMMANDS TO ACCOUNT FOR THE MORE C
 OMPACT PRINTING.#B#C-#LM10#LL60 AND NOW WE RETURN T
 O NORMAL MODE, IT IS VERY IMPORTANT TO DO THIS IF Y
 OU DO NOT WANT YOUR TEXT TO BE MISALIGNED WHEN USIN
 G DIFFERENT FONTS.#B2 FIG 3.#B1 THE NEXT DEMONDRATI
 ON IS OF INDENTED TEXT, A GOOD EXAMPLE OF THIS WOUL
 D BE TO INCORPORATE A QUOTE IN YOUR BODY OF TEXT:-#
 B1#LM20#LL40 GOOD WORD PROCESSORS SHOULD INCOROPRAT
 E EASE OF USE WITH FLEXIBILITY AND FULL CONTROL OVE
 R EVENTUAL TEXT LAYOUT ON PAPER!#B1#LM10#LL60 NOW W
 E RETURN TO THE MAIN BODY OF THE TEXT AGAIN, THIS C
 AN BE VERY USEFUL FOR MAKING PARTS OF THE TEXT STAN
 D OUT.#B2# FIG 4.TO PRODUCE A NEATLY ALIGNED LIST O
 F ITEMS WE NEED TO ENTER LITERAL MODE :-#LI#B1 1. J
 ETSET WILLY      - PLATFORM,           FIVE DRAGONS #B
  2. SPEED RACER       - GRAND PRIX,         FIVE DRAGO
 NS #B 3. WORLDS OF FLIGHT - FLIGHT SIMULATOR, FOUR
 DARGONS#B1#FJ NOW SEE WHAT HAPPENS IN JUSTIFY MODE:
 -#B1 1. JETSET WILLY        - PLATFORM,         FIVE D
 RAGONS #B 2. SPEED RACER          - GRAND PRIX,       F
 IVE DRAGONS #B 3. WORLDS OF FLIGHT - FLIGHT SIMULAT
 OR, FOUR DARGONS
  X
```

Not really a fault but an idea for a future product is a spell check routine to accompany EA, again on disc this could be a separate program which would load the EA text file check the spelling and formating errors, allowing it to be loaded back into EA for printing.

So how about it Wayne, why not correct the bugs, and take me up on my suggestion for your next product.

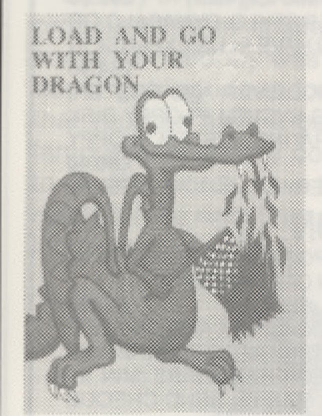
Program: *Electronic Author*
Supplier: Quickbeam
Price: £14.95 (Cassette) £19.90 (disc)

SPECIAL OFFER!

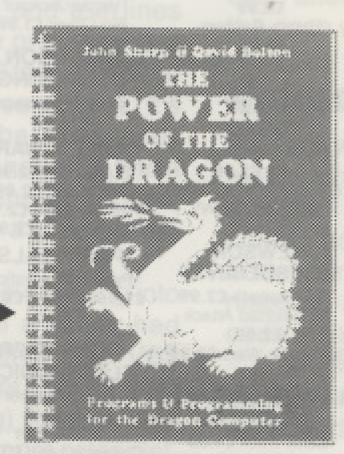
Exclusive to Dragon User readers

FREE GAME with first 450 orders

Dragon User and Abbott Computer Products have stocks of three well known Dragon books which are now no longer available from the original publishers. This means that we can offer them to readers at much reduced prices while stocks last.



◀ **Load and Go with your Dragon** by John Phipps and Trevor Toms. A popular starter's collection of programs with educational hints, a section on debugging, games, and a Wumpus. 128 pages. £1.95



The Power of the Dragon by John Sharp and David Bolton. A guide to programming the Dragon, undertaken through a collection of 30 entertaining programs from games to VAT calculation. 175 pages, £2.75 ▶

The first 450 orders will also receive a FREE GAME — Bonka is a ladders-and-platforms game for the Dragon 32. It's not one of the great DRAGON classics. It's not even new. But it's a great way to kill off those idle half hours when you feel like a break from reading or programming . . . and it's free.

These books have been travelling around looking for Dragon owners for some time, so although they are in perfect nick inside, most of the covers have a tiny dent, or sticker mark, or other small token of life's experience. These, like the low prices, are all part of the natural beauty of cut-price books, and do not affect their performance.

Dragon Magic — Your first programming book by Richard Wadman. This is a programming primer, approved as a Dragon teaching book by Dragon Data, ideal for youngsters (or not so youngsters) starting out. With line and colour illustrations. 56 pages, £1.50 ▶

Please send me the following DRAGON books:

. . . . LOAD AND GO, £1.95 each . . . . .

. . . . . THE POWER OF THE DRAGON, £2.75 each . . . . .

. . . . DRAGON MAGIC, £1.50 each . . . . .

TOTAL . . . . .

NAME . . . . ADDRESS . . . .

. . . . POSTCODE . . . .

Please make out cheques and postal orders to Scot Press Ltd. (Dragon books)

Bank account empty? Don't know where the money goes? Can't afford an accountant? Or even a lock-up wallet?

You should have DRAGON HOME ACCOUNTS for the Dragon 32.

- A 4pp booklet describes the options for storing your monthly household transactions on tape, making forward estimates of expenditure and reviewing your budget to date or month by month.
- The accompanying program was originally marketed by Gemini Marketing at £19.19. DRAGON USER's special offer price is £6.50, INCLUSIVE of postage and packing. The program comes on cassette only — we regret there is no disc version available.
- DRAGON HOME ACCOUNTS can handle up to 150 transactions a month of up to £10,000 each. The program can display a bar graph plot of monthly expenditure on any particular item, and store fixed values for regular monthly outgoings. Information from earlier months stored on separate tapes can be loaded into the program.

Please send me . . . copies of **DRAGON HOME ACCOUNTS**, at £6.60 each. **TOTAL** . . . . .

NAME . . . .

ADDRESS . . . .

Please make out cheques or postal orders to Scot Press Ltd. (Home Accounts)

IMPORTANT: When ordering special offer items, you can send your orders for BOOKS and for HOME ACCOUNTS to us in the same envelope if you wish, but please make out payments for BOOKS and for HOME ACCOUNTS on separate cheques, and send them with the separate coupons provided (photocopy coupons are acceptable). This helps us to keep the flow of orders separate, and send the goods out more quickly.

Send your orders to Dragon User at 12/13 Little Newport St., London WC2H 7PP, marking the envelope DRAGON SPECIAL OFFERS.

CLASSIFIED ADS

DRAGON 32. Konix Joystick, light pen, over 30 original games including Fire-Force, all leads, boxed £70 ono ask for Niaz Leave. Tel: 01-892 9985.

SWAP ALL, for printer with graphics or plotter (A4 min) or sell for £65. Dragon 32, dust cover, pair Delta Analogue joysticks, software incl. World of Flight, Telewriter graphics system, etc. Mags/books. All excellent condition. Tel: 02214 60552.

DRAGON NEWS magazine out now! Includes offers, games, reviews, software exchange, listings, adventure help and more! For your trial issue, send 50p & stamp to Dragon News, C. Vollbracht, 110 Alpine Rise, Styvechale Grange Estate, Coventry, CV3 6NR.

WANTED Race Sideways Rom or another Andtek Addbus PCB or complete. Will consider even if rest of system has to come with it. Cash waiting (0909 477734).

DRAGON 32 joysticks, plus £100's of games (all originals). Only £80. Call Ross on Chandlers Ford (0703) 260488. Any time.

DRAGON WITH NEW full spec 64 board p.s.u. leads. £45.00. Working 32 board £20.00. Offers please for edit + 32/64. Edit + DragonPlus 128 cartridges. Trojan light pen. Sprint compiler. Plus other utilities. Colwyn Bay (0492) 47813.

DRAGON GAMES for sale. Includes some latest titles and many Microdeal hits. Swapping considered — send details: for our full list send an s.a.e. to: Chris Vollbracht, 110 Alpine Rise, Styvechale Grange Estate, Coventry, CV3 6NR.

SUPERDOS disk drive controller kit (excluding case) £55: tested working Dragon 32 (no lid) £38: Untested complete could be working! £35: Most Dragon spares inc 74LS783 — £10.25, 6809E, 4164, transformers, CPU's, PSU'PCB. Phone for details — Nick Spiers 0892 44070 eve's.

OVER 100 GAMES for sale from 50p each + p&p. Also back issues of D.U. send s.a.e. to H. Grey, Three Trees, Green Farm Lane, Billericay, Essex.

DRAGON 32, joysticks, £100's worth software, over forty Dragon users — £120. Telephone (091) 5220727 evenings.

DRAGON computer & games, tape recorder, joystick, manual. Can be split. Tel: Harby (0949) 60451.

HERE'S MY CLASSIFIED AD.
(please write your copy in capitals on the lines below)

Name . . . .
Address . . . .
. . . . **Tel:** . . . .

Please cut out and send this form to: Classified Department, Dragon User, 12–13 Little Newport St, London WC2H 7PP.

Write: ADVENTURE

Peter Gerrard talks about words and anniversaries

AS promised last month, we'll be spending some time examining a more detailed version of our parser. We might not have a parser that compares with *Magnetic Scrolls*, but if we produce any hints and clues for the finished game at least we'll be able to type them all in! (Severe bug in at least one computer version of *The Pawn*).

By way of a slight diversion, after all the programming is finished for the month, someone recently was asking how to go about getting ideas for an adventure so that you can have a game ready when a famous anniversary comes around. How to, in other words, have the right adventure ready for the right time and thus hopefully increase your sales.



But first, the programming, and as a look at the listing will tell you, what a lot it is too.

Some things, you'll be pleased to note, remain pretty much the same as before. In particular, the routine to accept an input from the player hasn't changed by very much. Lines 60000 to 60032 take care of all this, and if you managed to understand the explanation in the June issue then you'll have no trouble in working your way through this lot. The only real difference are a) we keep track of the previous entry from the player in case they want to repeat a command and b) we allow for keys like quotation marks (for talking to other characters) and commas, which are used for separating commands.

This means that we can happily accept inputs of the form:

SAY TO GEORGE "ATTACK THE TROLL" SMASH THE DOOR,AGAIN,WALK THROUGH,GO SOUTH,EXAMINE THE GROUND

or whatever turns you on, I suppose.

Now all we have to worry about is the code that makes sense of all this lot. As with accepting the input, some of it will be quite familiar, and it all really works on the same principle as the mini VERB ... NOUN parser covered in the June issue. If you read the July one you should have got most of your vocabulary sorted out by now, or at least all the essential parts of it, but the program includes just a few words to enable us to test you've entered this month's section of program correctly.

We'll take each new addition in turn, and concentrate first of all on the section that looks for not only verbs and nouns, but also link words, action words, and additional nouns (if you'd typed in something like PLACE KEY UNDER DUSTY BOOK).

Lines 4016 to 4042 split the sentence up as appropriate, taking out spaces and references to the words 'the' and 'a' in lines 4020 to 4026. Note that the presence of a verb is checked in lines 4016 to 4018 before we start removing parts of sentences. It's no use chucking out all the extraneous words if we haven't even managed to find a single space in the player's input. A space, after all, is what separates the verb from the others words.

Line 4044 then truncates everything into a five letter word, so that if a player had entered as his verb EXAMINE then VB$ would just contain the letters EXAMI after passing (or is that parsing) through this line. If you want to check for more or less letters then just alter this line accordingly.

Lines 4046 to 4074 then do all the checking for known verbs, link words, nouns and so on. Variables like VB, NB and so on are returned as being the verb number, the noun number (LI being the link word number and AC the action word number: nouns are split into ordinary nouns and additional nouns, and have the variables NB and NA attached to them.)

Assuming everything is found to be a word that the program has been taught to recognise, the program trundles through to line 4084 and returns from this section of the game. Now, having got numbers to represent verbs, nouns, and so on, you can easily include code of the sort:

```
ON VB GOTO 1000,1050,1100
```

and so on, so that verb one would take us to line 1000. Here, if you use CP to represent the player's current position in the game, we could have something like:

```
1000 IF CP=1 AND L1=13 AND NO=21 AND OB%(LI)=-1 THEN PRINT "You throw the book at the king in disgust and storm out.":OB%(LI)=CP:ZZ=ZZ-1: RETURN
```

Here we have assumed that the thirteenth link word is a book, the twenty first noun is the king, the first verb is the word 'throw' and OB% contains the position of all the objects (minus one indicating that it's being carried) while ZZ is the total number of objects that the player is carrying at the time. Thus the book is thrown, it is placed in the room (by the statement OB%(LI)=CP) and the number of objects being carried is decreased by one. All this in response to the simple sentence from the player:

THROW THE BOOK AT THE KING

What if the player wasn't carrying the book? Well, line 1002 could read:

```
1002 IF CP=1 AND LI=13 AND NO=21 AND OB%(LI)<>-1 THEN PRINT'But you aren't carrying the book.':RETURN
```

Perhaps you'd like to expand on the response and have the king set about beating up the player for his impudence in throwing something at a mighty ruler. Or even attempting to throw something, as is the case with line 1002.

Unknown words

All very well if the program understands every word in every known tongue, but alas there will come a time when the program comes across something that it doesn't understand or which, regrettably, it recognises as a bad word. So, we have some code to take care of that.

Lines 4076 and, more importantly, 4012 to 4114, look after the latter of these options, and I'm sure that should wholesome people as readers of *Dragon User* cannot begin to imagine what rude words are being replaced by asterisks here. This is a family magazine, come on, we don't want to get into trouble. Freddie Starr ate my dwarf indeed.

Words that aren't recognised are covered in lines 4080 to 4090, where DN$ contains the word that the program doesn't know. Okay, so it should be DK$, we can't all be perfect. DN$ itself can come from a variety of different places, including lines 4048, 4054, and so on. Anywhere, in short, where an unknown word can be found.

The little routine in lines 4086 to 4090 comes into play if the player is attempting to use a verb as a noun, or vice versa perhaps, and tells the player that the word is a known one but isn't recognised in the context that he is trying to use it in. Otherwise, he just gets told that the word is an unknown one.

Repeated instructions

By using the word AGAIN, or more simply the abbreviation A, a player can get the computer to repeat the last instruction. Unless, of course, the last instruction contained an unknown word, in which case line 4080 might come into play. Assuming that all has been understood (including something like OPEN DOOR,CLOSE IT, where the 'it' is translated back to be a 'door' again by line 4064) then lines 4006 and 4008 look first of all to see if the first thing encountered in a sentence is an AGAIN or an A, and if it is, then sort out the input strings accordingly. Now you know why the input routine was modified slightly so as to keep track of the previous entry by the player.

Lines 4092 to 4100 are also used by the AGAIN routine, but only if something is found that can be AGAINed, if you'll excuse the phrase. Here we are also look-

ing for additional commands, separated by the use of a comma. You'll note the search for this in line 4096, so that the program can be persuaded to cope with things like:

PUT KEY IN DOOR,OPEN IT,GO SOUTH,A,A,TAKE AXE FROM WALL.

And if you find that the program doesn't work then I shall use the excuse that *Magnetic Scrolls* put forward when writing to a friend of mine after he had complained about a fault in one of their programs.

The program worked perfectly, they told him, it was the design of the computer that was at fault!

No, seriously, it does work.

Conclusion

Amazing how rapidly one can run out of space. I was well prepared to talk all about getting ideas for adventures, searching through encyclopaedias for dates of historical importance, and so on. After all, it is quite important if you can manage to produce the right adventure at the right time.

Whoever gets something out by January of next year, in time for the 200th anniversary of all future England cricketers being shipped to Australia, will make a bob or two, provided that the adventure is any good of course!

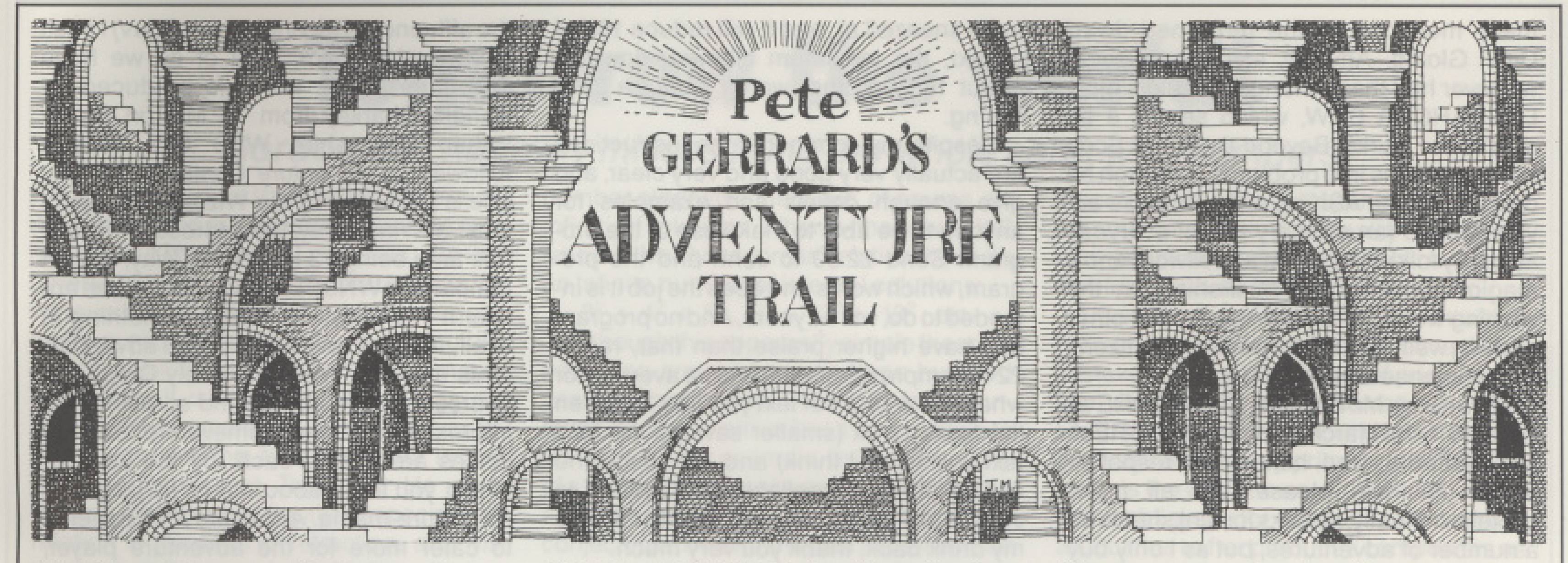
But, as a brief word, don't always go for the blatantly obvious. If it is that obvious, a bigger company than you or I will be producing something, you can bet your life on that. Don't always go for exact anniversaries either, as I think we can safely assume that historians do not know exactly to the second when some of the more ancient events took place. Give them (and you!) a year or so leeway in either direction. Finally, look for anniversaries that you can exploit and perhaps get some educational sales out of, apart from any others you might gain.

The Crusades, perhaps. The Third Crusade (check your dates!). We'll return to this topic at a later date, methinks.

For now, if you can enter the parser in readiness for next month's issue that will be sufficient. All homework excused.

```
Parsing by ...

4 x$=CHR$(34):dw$="I don't know the word
 "
5 DIM vb$(50),ac$(50),li$(50),no$(50)
6 nb$(1)="door":vb$(2)="close":na$(1)="d
oor":nb$(1)="smaller":vb$(1)="open":li$(
1)="blue":ac$(1)="on"
7 nz=1:nv=2:nn=1:nc=1:nl=1
10 GOSUB 4000:PRINT "vb$="vb$:PRINT"ac$=
"ac$:PRINT"li$="li$:PRINT"na$="na$:PRINT
"nb$="nb$:GOTO 10
4000 REM inder of something
4002 pr$=cm$:pn$=na$:pp$=n1$:se=0:ag=0:I
F an<>0 THEN cm$=an$
4004 PRINT:PRINT ">";:GOSUB 60000
4006 IF LEFT$(cm$,5)="again" THEN ag=1:c
m$=pr$:an$=MID$(an$,7):pk$=MID$(pk$,7):I
F an$="" THEN an=0
4008 IF LEFT$(cm$,1)="a" THEN ag=1:cm$=p
r$:an$=MID$(an$,3):pk$=MID$(pk$,3):IF an
$="" THEN an=0
4010 na$="":nb$="":li$="":ac$="":vb$="":
na=0:nb=0:vb=0:ac=0:li=0:v1$="":l1$="":a
1$="":na1$="":nb1$=""
4012 lc=LEN(cm$)
4014 GOTO 4092
4016 FOR i=1 TO lc:IF MID$(cm$,i,1)=" "T
HEN v1$=LEFT$(cm$,i-1):k=i:GOTO 4020
4018 NEXT i:v1$=cm$:GOTO 4044
4020 FOR i=k TO lc:IF MID$(cm$,i,3)=" a
" THEN cm$=LEFT$(cm$,i-1)+MID$(cm$,i+2)
4022 NEXT:FOR i=k TO lc:z$=MID$(cm$,i,5)
4024 IF z$=" the "THEN cm$=LEFT$(cm$,i-1
)+MID$(cm$,i+4)
4026 NEXT
4028 FOR i=k+1 TO lc:IF MID$(cm$,i,1)="
"THEN l1$=MID$(cm$,k+1,i-k-1):k=i:GOTO 4
032
4030 NEXT:na1$=MID$(cm$,k+1):GOTO 4044
4032 FOR i=k+1 TO lc:IF MID$(cm$,i,1)="
"THEN a1$=MID$(cm$,k+1,i-k-1):k=i:GOTO 4
036
4034 NEXT:na1$=MID$(cm$,k+1):GOTO 4044
4036 FOR i=k+1 TO lc:IF MID$(cm$,i,1)="
"THEN na1$=MID$(cm$,k+1,i-k-1):k=i:GOTO
4040
4038 NEXT:na1$=MID$(cm$,k+1):GOTO 4044
4040 FOR i=k+1 TO lc:IF MID$(cm$,i,1)="
"THEN nb1$=MID$(cm$,k+1,i-k-1):k=i:GOTO
4044
4042 NEXT:nb1$=MID$(cm$,k+1)
4044 nb$=LEFT$(nb1$,5):na$=LEFT$(na1$,5)
:ac$=LEFT$(a1$,5):li$=LEFT$(l1$,5):vb$=L
EFT$(v1$,5)
4046 FOR i=1 TQ nv:IF vb$(i)=vb$ THEN vb
=i:GOTO 4050
4048 NEXT i:vb=0:dn$=v1$:GOTO 4076
4050 IF li$="" THEN 4056
4052 FOR i=1 TO nl:IF li$(i)=li$ THEN li
=i:GOTO 4056
4054 NEXT:li=0:dn$=l1$:GOTO 4076
4056 IF ac$="" THEN 4062
4058 FOR i=1 TO nc:IF ac$(i)=ac$ THEN ac
=i:GOTO 4062
4060 NEXT:ac=0:dn$=a1$:GOTO 4076
4062 IF na$="" THEN na=0:RETURN
4064 IF na$="it" THEN na$=pn$:na1$=pp$
4066 FOR i=1 TO nn:IF na$(i)=na$ THEN na
=i:GOTO 4084
4068 NEXT i:na=0:dn$=na1$:GOTO 4076
4070 IF nb1$="" THEN 4076
4072 nb$=LEFT$(nb1$,5):FOR i=1 TO nz:IF
nb$(i)=nb$ THEN nb=i:GOTO 4084
4074 NEXT:nb=0:dn$=nb1$:GOTO 4076
4076 se=1:IF vb$="****" OR vb$="****" TH
EN PRINT "Watch it.":RETURN
4078 GOTO 4086
4080 IF dn$="a" OR dn$="again" THEN PRIN
T"You're not making much sense I'm afrai
d.":dn=1:se=1:RETURN
4082 PRINT dw$;x$;dn$;x$;".":dn=1
4084 RETURN
4086 v$=LEFT$(dn$,5):FOR i=1 TO 50:IF v$
=vb$(i) OR v$=ac$(i) OR v$=li$(i) OR v$=
no$(i) THEN 4090
4088 NEXT:GOTO 4080
4090 PRINT dw$;x$;dn$;x$;" in that conte
xt.":RETURN
4092 IF ag=1 THEN 4102
4094 an=0:an$="":FOR i=1 TO lc
4096 IF MID$(pk$,i,1)="," THEN an$=MID$(
pk$,i+1):cm$=LEFT$(cm$,i-1):pk$=MID$(pk$
,i+1):an=1:GOTO 4102
4098 NEXT
4100 IF an$="a" OR an$="again" THEN an$=
cm$
4102 lc=LEN(cm$):se=0:FOR i=1 TO lc
4104 IF MID$(cm$,i,7)="*******" THEN cm$
=LEFT$(cm$,i-1)+MID$(cm$,i+8):GOTO 4114
4106 IF cm$="********" OR cm$="********"
 THEN PRINT "And you.":se=1:RETURN
4108 t$=MID$(cm$,i,8):IF t$="********" O
R t$="********" THEN PRINT"Behave.":GOTO
 4124
4110 IF MID$(cm$,i,4)="****" THEN PRINT
"Curb your tongue.":GOTO 4124
4112 NEXT:GOTO 4120
4114 PRINT"Stop swearing.":GOTO 4124
4116 RETURN
4118 PRINT "Erm. no.":se=1:RETURN
4120 REM inder of something
4122 GOTO 4016
4124 se=1:RETURN
60000 p1$=cm$
60002 cm$="":h=1
60004 PRINT "+";CHR$(8);
60006 a$=INKEY$:IF a$="" THEN 60006
60008 REM
60010 REM inder of something ELSE
60012 z=ASC(a$)
60014 IF (z>97 OR z<65) AND z<>32 AND z<
>44 AND z<>13 AND z<>8 AND z<>34 THEN 60
006
60016 zl=LEN(cm$):IF zl>75 AND (z<>13 AN
D z<>8) THEN 60006
60018 IF z=32 AND zl=0 THEN 60006
60020 IF z=44 AND zl=0 THEN 60006
60022 IF z=13 AND zl<>0 THEN PRINT " ";C
HR$(8);".":pk$=cm$:RETURN
60024 IF z=13 THEN 60006
60026 IF z=8 AND zl<>0 THEN cm$=LEFT$(cm
$,zl-1):PRINT a$;:GOTO 60004
60028 IF z=8 THEN 60006
60030 IF z=34 AND zl=0 THEN 60006
60032 cm$=cm$+a$:PRINT a$;:GOTO 60004
```



WHAT a quiet unassuming lot you Dragon adventure fanatics are. Apart from British Telecom engineers with Dragons who use British Telecom stamped envelopes to get in touch, and people who write on pink paper, or yellow paper, or (urgh!) green paper, something like ninety per cent of the letters I get are grovelling pleas for help with various adventures. Of the rest about half are offering solutions to vast numbers of different games, and the other half are wittering on about *Jet Set Willy* and other things only obliquely related to the business of solving adventures. Something strikes me as odd here, somehow. Does it mean that nine out of ten Dragon owners, who expressed a preference, are incapable of solving adventures? I shouldn't think so. Does it mean that nine out of ten Dragon owners are like the writer of this column and are a wee bit on the lazy side? Ahah, much more like it, so let's have some more solutions please, chaps.

Solving everything

Onto someone who has probably solved more Dragon Adventure games than have been written, and is now going about writing his own just to redress the balance. Young Hargrave, for it is he: a letter from him is like a stroll through a post office. Wads of paper, jiffy bags, stamps, more envelopes, tapes, all sorts of interesting things. Plus, and this is what they want, solutions. For 20 pence, to cover printing costs (it says here) he can offer solutions to *Tanglewood*, *Vortex Factor*, *Colossal Cave*, *Trekboer*, *Aquanaut 471*, *E1 Diablero*, *Black Sanctum*, *Madness and the Minotaur*, *Juxtaposition*, *Don't Panic*, *Ket* (part one), *Syzygy*, *Starcrash*, and something called others! By which of course he means every other adventure in the known universe, I have no doubt. His address is Crawley Hill Farm, Uley, Dursley, Gloucester GL11 SBH, although that could be 5BH I have to admit. Simon Hargrave, may your postman curse this column.

The game he sent to me, *Starcrash*, was mentioned by big brother in the January column, so we won't go into any great detail about it here, except to say that it had been updated somewhat since then. It includes a very useful verb, VLIST, which prints out a list of all known verbs. Very handy when you're convinced that the program writer has been watching too many episodes of **Call My Bluff**. While trundling about an unknown planet awash with alien life forms and acid , which is a bit like going into Watney's pub in Birmingham, I was reading Simon's letter accompanying letter, or script for the next six months of Eastenders, whichever is the shorter. He is trying to market this game, but says that the software houses will not themselves take on the responsibilities of marketing a Basic (with machine code additions of course) adventure.

Now the software houses can naturally enough please themselves about what they decide to take on or ignore, but Simon's point is worth pursuing. Most of the retailers will not stock any adventure that doesn't contain graphics, unless it's by Infocom. Which is not, by the way, a complaint about Infocom. Good luck to them. The excuse that most of these retailers give is exactly the same one as Simon gets when trying to market his Basic adventure, namely that there's no demand for them. Of course there's no demand for them if nobody stocks the blessed things! A butcher who didn't stock meat would probably come to the conclusion that there was no demand for it because no-one came into his shop. How do they know? Market research on people who come into the shop? If you know you're only going to be able to buy over-priced and over-hyped (generally) graphic adventures and you don't like graphic adventures then you won't go into the shop, and so the results of any market research done in that shop will be totally incorrect.

What can be done about this sort of situation? After all, if W.H. Smiths won't stock a game you can hardly persuade them to do so. If Ostrich Software won't sell your game you can't tell them that they must on penalty of death to the Spanish Inquisition. After all, no-one expects etc. etc. etc. Personally I liked Simon's *Starcrash*, but haven't got the clout to market it. I can't rush around Wigan (of which more later) throwing copies into every shop I come across. Perhaps you'd like to tell me your views on the subject. Have you tried marketing a game of your own? What sort of problems have you had, and more importantly, did you find any solutions to them? Simon Hargrave, and I await your letters.

Back to this letter of the young boy Hargraves again. He offers us some POKEs for *Keys of the Wizard*, if you're having problems with this particular bash. Three little goodies in all, and they are:

POKE &H200F,&H12 and
POKE &H2010,&H12.

This disables death, which would be most convenient for a lot of us I imagine. According to Simon this takes care of most death situations, and all you have to do is press 'W' to re-incarnate yourself. I shall try and remember that next time I visit the Seven Stars.

POKE &H1E9E,0-255.

This governs the strength of all the creatures in the game. However, set it too low and you'll manage to stop the Unicorn helping you, which is definitely not what they want.

POKE &H181C,&H0F.

This allows you to rest whenever and wherever you want, so that you can get your strength back whenever it's needed.

Unfortunately, even with the help of this lot Simon hasn't managed to finish the game, and seems to be convinced that it's impossible. As if.

One more mention for him, and then we'll ignore him for the rest of this month, and possibly next month as well. Not that I think he'll go away.

Tips on *Tanglewood*, and no backwards writing because all the blood rushes to my head. To reach the fireflies you must go north, east, west, north with Foghorn at the southern entrance to the marsh at night.

To get Bruce to the island you must drink in the Western Well and then jump on the springboard.

To get Beanbag into Schark Castle you'll have to wait until the end of the game. You can enter by using the fire exit, but only if you can find "the silly thing!" as the Hargrave pen has it. He also doesn't mention where it is, gloating about people who can't find it, talking about certain animals who like to bury their heads in the sand ...yawaevigatahwoh!

Northwards

Enough of him, let's leap about the countryside and head further north even than Wigan. Heavens. John Beagrie, up in Aberdeen, tells us that he had managed to sort out the Gi-Ants, who seem to have appeared in almost as many games as the legendary Grues, and who pick up as

many mentions as the renowned dwarf Dimli Gloing. Anyway, to get past them whoever had the archangel should simply LEGNAHCRA DEW, which seems a bit drastic but works. Beyond there our Scottish friend runs into problems, although he does tell us the ASKing at the citizen's advice bureau (an ordinary tale of everyday advisory folk) will get you a spellword and a magic carpet, and that wishing on the wishing well will transport you to the other wishing well, next to old friends the citizens who dispense advice.

By the way, Mr. Beagrie, the verb BRE as covered in the March issue refers to BREak not BREathe, and hence the response about violence, please. We all make mistakes. Finally, he asks for hint sheets on a number of adventures, but as I only buy sheets of paper in units of 500 at a time you'll have to wait a wee while.

Text compressor

Continuing our countryside forays and cunningly heading ever closer to Wigan we get to Sheffield, wherein lives John Foster, at 94 The Oval (personal cricket ground, must be well off), Firth Park, Sheffield S5 6SP. Telling me about large friends of yours does not guarantee a mention, John, especially when they're called Arnold and Sylvester. The latter, as we all know, is about as tall as the aforementioned Dimli Gloing, but I daresay if Arnold (no I can't spell his surname!) spilt my drink I'd probably buy him another one. John has written a Basic text compressor for Dragon adventure writers, which costs £2.00 plus 50 pence post and packaging, and this magnum opus he refers to as System One.

The instructions for use are about as long as a good issue of *Dragon User*, but it isn't that difficult to use. We at Chez Gerrard managed quite successfully, even under the strain of trying to watch Ian Botham smash several cricket balls out of the ground. To work at all you'll need to have at least 2K of text in your game, which shouldn't be any great strain for most of us, and although it does take quite a while to hunt through everything and compress it down in size, if you haven't got access to a routine like this then it's worth the wait. Not the greatest saving of memory that I've ever seen , although any saving at all is not to be sneezed at, and it will reduce 18K of of text, the maximum allowed, down to about 14K. In other words, about a 22% saving.

Despite my comments the instructions are actually very good and very clear, and give enough details and examples for anyone to be able to make use of the program. Send £2.50 to John and the program, which works and does the job it is intended to do, will be yours, And no program can have higher praise than that, really, 22% compression seems to be average for whatever amount of text you use, between the limits of 2k (smaller savings the less text you have, I think) and 18K. So, send Arnold of the unspellable surname off to John and if it reduces him by 22% I'll have my drink back, thank you very much.

Whoever started off this desktop publishing lark has a lot to answer for. Whoever coined the phrase has even more to answer for, since the idea of reading the tops of desks is not one that has ever appealed to me personally. Still, it has prompted a lot of people to start producing magazines of varying quality, and two more for all you adventure freaks to start sending off for are both produced in (pause for fanfare, blare of trumpets, Land of Hope and Glory etc.) Wigan.

One of the advantages of living in Wigan (yes, there is more than one) is that one can get to meet the producers of these magazines, and despite the fact that one of them beat me at pool (I was cueing under the affluence of incohol, honest guv) I shall still mention them. First of all we have *Adventure Probe*, which is produced by Sandra Sharkey from 78 Merton Road, Wigan Lancashire WN3 6AT, closely followed by *Adventure Contact*, and this one is produced by the Wizard of the pool cue Pat Winstanley, the address to write to this time being 13 Hollington Way, Wigan Lancashire WN3 6LS. Both magazines are over a year old, which says something in itself, and both are full of useful adventure hints and tips. Not exclusively Dragon of course, but you should find something to interest you in there somewhere. Sample issues are a quid each, by the way, for which you'll get about 40 or so pages of adventure-mania. *Adventure Probe* seems to cater more for the adventure player, while *Adventure Contact* tends to look after the adventure writers amongst us, so take your pick. Those areas are not, however, mutually exclusive. If I ever win a game of pool against one or the other of them I shall let you know.

Oh well, out of space again, so I'll leave you with something to ponder on until next time. We're all used to collective terms for things, like a flock of sheep, a herd of cows, a crowd of people, or sillier ones like a column of accountants, an abbreviation of estate agents and a box of matches. But what would you call a collection of adventure enthusiasts? A text of adventurers, a problem of adventurers, a scroll of adventurers? What do you think?

Adventure Contact

To help puzzled adventurers further, we are instituting an Adventure Helpline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpine, 12/13 Little Newport Street, London WC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Trial to write to as well!

Adventure . . . .

Problem . . . .

. . . .

Name . . . .

Address . . . .

. . . .

Adventure Contact

Adventure: *Poseidon*
Problem: How to get across the Pool of Fire on the 3rd level.
Name: D. Beckwith
Address: 30, Rachdall Road, Leyton, London E17 8JF.

Adventure: *The Pawn*
Problem: How do I get light, so that I can see something inside the tree?
Name: Henning Johnson
Address: Gamle Sandrigvei 20, 4816 Kolbsornsvik, Norway.

Adventure: 1) *Vortex Factor* 2) *Juxtaposition*
Problem: 1) I've done everything but haven't finished it 2) Where's the oil for the rusty droid?
Name: James Wood
Address: 49 Rockwood Crescent, Woodhall, Pudsey, W. Yorks LS28 5AD.

Adventure: *Tanglewood/ Juxtaposition*
Problem: How do you get the fishing rod? How do you get the lube tube in the hydrophonics dome?
Name: Neil Davies
Address: 4 Grinshill Drive, Crowmoor, Shrewsbury, Shropshire SY2 5JE.

Adventure: *Shenanigans*
Problem: How can I exit from the apartment? What does the landlord want? Where is this? Help!
Name: Dario Palmieri
Address: 110 Molino 101 Pescarola 72, 40131 Bologna, Italy

Adventure: *The Larkspur Trilogy*
Problem: How do I open the cabinet? What do I do at the Idol's Shrine to get the saw in the statue's hand? How do I get one paddle from the humming generator without being electrocuted? Help!
Name: Mr. S. Albert
Address: 22 Littledown Orchard, Newton Poppleton, Sidmouth, Devon.

Adventure: *Airball*
Problem: Can I pick up more than one crate at a time, if so, how?
Name: Nicholas Cutts
Address: 88 Hale Lane, Mill Hill, London NW7 3RT

Tomorrow the world!

Gordon Lee narrowly misses filling the whole earth with numbers

PIERRE de Fermat (1601-1665) was one of the two leading mathematicians of the first half of the 17th century (Descartes was the other). His wide ranging discoveries in various branches of mathematics — geometry,number theory, differential calculus, and probability — are all the more remarkable when it is realised that he was an amateur mathematician, being a lawyer by profession, serving as a councillor in the local parliament at Toulouse. He also studied the science of optics, especially in relation to the laws of refraction, and his theorem on the speed of light, though conflicting with the views held at that time, was later shown to be correct — but not until after the passing of two centuries!

Many of his mathematical theorems were also not proven until long after Fermat's death. one of these, known as Fermat's 'last' theorem is still unproven to this day. This theorem states that the expression

$$x^n + y^n = z^n$$

has no whole number solution when n is greater than 2. In other words there are, for example, no two cubes which sum to a third cube, no two 4th powers which sum to another 4th power, and so on. Remember that each of the values must be a whole number otherwise there are, of course, an infinity of solutions. When n (the power) is 2 we have the 'Pythagorean' relationship, with an infinite number of integral solutions:

$$3^2 + 4^2 + 5^2 ,\ 5^2 + 12^2 = 13^2 \text{, etc.}$$

Although this apparently simple theorem has been around for over three hundred years, no one has yet been able to demonstrate its proof, neither has it been shown to be wrong — for example, by the discovery of a single instance of a solution containing a higher power. As with other of Fermat's conjectures, he evidently felt intuitively that they were correct, found plenty of examples to support this theorem, and was unable to find any examples which disproved them.

A number of his other conjectures were in relation to the infuriating class of numbers known as primes. One of these theorems states that every prime number of the form 4n + 1 (eg 5, 13, 17, 29, 37, 41, etc.) is expressible as the sum of two squares in one way, and one way only.

Another states that if p is a prime number and a is any positive integer, then ap — a is exactly divisible by p. If both of these theorems are correct then they could be useful tools in the search for the higher primes. In fact the proofs for both of these theorems has been established, but again, not until after Fermat's death. Two of the mathematicians who worked on them were Gottfried Leibniz and Leonhard Euler.

However, there is one instance in which Fermat has been shown to have been mistaken. Since the impact of prime numbers was first noticed, mathematicians have sought (in vain) for a formula which will generate exclusively prime numbers. One such, is the well known x2 + x + 41, which works obligingly for values of x from zero to 39, but thereafter produces composites as well. Fermat, while studying the possibility of such formulae, came up with one of his own which, he maintained, produced only prime numbers. This formula was:

$$2^{2^n} + 1$$

That is, 2 raised to the power of 2n, plus 1, where n is any positive whole number (including zero)

The following table gives the first nine values:

n=				
0	=	2^1+1	=	3 (prime)
1	=	2^2+1	=	5 (prime)
2	=	2^4+1	=	17 (prime)
3	=	2^8+1	=	257 (prime)
4	=	$2^{16}+1$	=	65537 (prime)
5	=	$2^{32}+1$	=	4294967297
6	=	$2^{64}+1$	=	(a 20-digit number)
7	=	$2^{128}+1$	=	(a 39-digit number)
8	=	$2^{256}+1$	=	(a 78-digit number)

To be fair to Fermat, he only stated a belief in his formula since the calculations needed to establish the larger numbers in the series were not possible at that time. The numbers formed when n is equal to 0,1,2,3, and 4 were easily shown to be prime, and so Fermat stated that it was probable that the rest of the series were also, including the number 4294967297, found when n=5. In fact, this number is composite and has a factor of 641. It is surprising that he was unable to determine this factor, especially as there is record of him factorising much more difficult numbers such as 100895598169 into its two prime factors 898423 and 112303.

It has now been proved that all values of n that are greater tham 4, produce numbers that must be composite — almost the exact opposite of Fermat's original theorem! The formula still resents a challenge to some mathematicians to discover factors of some of the larger numbers in the series, and one of the largest numbers for which a factor has been found is when n=73. The number itself has over 3.10^{21} digits (that is 3 followed by 21 zeros), but one of its factors has been shown to be:

$$2^{75} \times 5 + 1$$

In everyday terms, this factor is equal to 2 raised to the 75th power.This number is then multiplied by 5, and 1 is added to the result.

The competition this month is to devise a program which computes exactly the value of this factor. NB Don't try to compute the actual number that this is the factor to! The whole earth is not large enough to contain it!

Prize

NOW that summer is with us (and if it isn't with us by the time you read this, I shall probably have emigrated to Australia), what would be a more ideal prize than an elegant, outsize t-shirt, proclaiming to all and sundry that one is a Dragon? Peaksoft are giving us twenty of their Dragon t-shirts as prizes for the August competition, so this will be the first *Dragon User* competition *ever* to ask you to state your size as well as your solution. Small, medium, large or extra large. You have one already? Get another one for the partner/toddler/tortoise. Let them see where you stand.

Rules

Firstly, do as we suggest. Stick to calculating the factor and not the actual number. While *Dragon User* makes every effort to ensure that the contents of its articles are accurate, it cannot be held responsible for the end of the world as we know it.

Secondly, send us your solution, any notes you want to include, a printout of your program (no tapes, please0, and your name and address in an envelope marked AUGUST COMPETITION.

This month, not so much a tie breaker as a tie-die breaker. Tell us what you will be doing with the t-shirt if you win it. It doesn't have to be true, only imaginative. And printable.

May winners

The number of three-letter word sequences discovered by entrants varied hugely, from a couple to 'nine pages of words, and I haven't got time to go through all of them', but as the competition was to devise a program, not to read a dictionary, the actual number of sequences wasn't the crucial factor. However, thanks for all the words we've never heard of...

Copies of Pete Gerrard's *Exploring Adventures on the Dragon*, published by Duckworth, go to Richard Long of Camberley, C. Hitchman of Middlesbrough, T. Fawcett of Hendon, Terry Potter of Chiseldon, S.A. Siddiqui of Chiswick, Terry Smith of Harlow, E.A. Newman of Addlestone, Phil Sapiro of Liverpool, P. Morgan of Bristol, Paul Weedon of Wotton-Under-Edge, Jake Anderson of Edinburgh, D. Denman of Twickenham, R. Banks of Southampton, Keith David of Crawley, P.D.Maddocks of Maidenhead, A.J. Westwood of Northampton, Simon Hargrave of Dursley, John Crighton of Hampstead, Pete Faraday of Warrington and G.R.Barber of Sutton Coldfield.

The tie-breakers had a heavy cricketing bias, but we liked the chap who had actually done some research: "Consultation with the Internat. Duvet Manufacturers' Association and the RSPB (spokesperson G. Gooch Esq.) produced a difference of option. However, the oldest local inhabitant recalled a duck worth life itself (Somme, 1916)."

Solution

This month's solution appears on pages 6 and 7. So does last month's.

JOHN PENN DISCOUNT SOFTWARE

RADIO SHACK CARTRIDGES
FOR THE DRAGON AND TRS 80

BRIDGE TUTOR	£10.00
GRAPHICS PACK	£10.00
HANDY MAN	£5.00
ROMAN CHECKERS	£5.00

CASSETTE GAMES

FOOTBALL MANAGER (Addictive)	£4.00
NUMEROLOGY (Occult)	£5.00
THE 13TH. TASK (text adv.) (Arc)	£3.00
COLOSSAL ADVENTURE (text adv.) (Cowan)	£5.00
HARE RAISER double pack (gr. adv.) (Haresoft)	£4.00
SUNKEN CITY (gr. adv.) (Slik)	£4.00
PREDICTOR (Benley)	£4.00
TIM LOVES CRICKET (Peaksoft)	£4.00
SAS (Peaksoft)	£2.00
PHOTO FINISH (Peaksoft)	£2.00
BACK TRACK (Incentive)	£3.50
MANIC MINER (Software Projects)	£3.50
DOMINOES (Oasis)	£3.00

BARGAIN BASEMENT
£1.50 each; five for £6.00; ten for £10.00
Please give at least two alternatives if possible.
GOLF: CHOCOLATE FACTORY: CITY DEFENCE: UXB: DON'T PANIC: OSSIE: MINED OUT: NORTH SEA OIL: DETECTIVE: DRAGON TREK: SUPERSPY:SURPRISE: ALIEN ODESSY: STOCKMARKET: GRID RUNNER: NIGHT FLIGHT: MINIGAMES: PICTURE PUZZLE: CELLS & SERPENTS: DRAG RUNNER: FLIPPER: ULTIMATE: ADVENTURE: PETTIGREW'S DIARY: TRANSYLVANIAN TOWER: LEGGITT.

UTILITIES/BUSINESS SOFTWARE
OS9 programs for use with OS9 Operating System, disc drive(s) and Dragon 64.

DYNACALC, BASIC 09	£21.00 each
'C' COMPILER	£27.00
R.M.S.	£20.00
PASCAL, CASH BOOK & VAT, STOCK RECORDING	£19.00 each

10% discount if 3-4 programs ordered; 20% discount if more than 5 ordered.

MANUALS
Cash Book & VAT; Stock Recording: Pascal; Basic 09. £2.50 each

BASIC 42; disc (Harris) for use with 64.	£14.95
+ HELP UTILITY £5.00 + SPOOL UTILITY	£5.00
(Please state whether SuperDos or DragonDos)	
MONEYBOX: disc (Harris)	£13.50
SHAREBOX, MAILBOX, ORDERBOX, (Harris)	£15.50 each
SALESBOX, BILLSBOX, CASHBOX, STOCKBOX (Harris)	£18.50 each
(SuperDos or DragonDos)	
SOURCEMAKER (Pamcomms) cass.	£8.00
DISK-KIT (Pamcomms)	£10.00
HI-RES-TEXT (Cartwright)	£3.00
SHAPER (Shards) — sound utility	£3.00
PERSONAL BANKING SYSTEM (Hilton)	£9.95
UTILITIES 1, DISASSEMBLER, (Hilton)	£5.00 each
MUSIC MAKER (Maken)	£5.00
MAILER/ADDRESS BOOK (MST) cass.	£2.50
STOCK CONTROL (MST) cass.	£2.50

EDUCATIONAL CORNER

Dragon Data series : NUMBER PUZZLER : CIRCUS ADVENTURE.	£2.00 each or two for £3.00
Shards/Cambrian/Tiger software TIGER GRAND PRIX , FUN TO LEARN, FAMILY PROGRAMS, MONSTER MATHS, SNOW QUEEN, QUIZ PACK : INFANT PACK.	£1.50 each or three for £3.00
Ampalsoft (Cheshire Cat series) MATHS 'O' LEVEL : MATHS LEVEL I (4-6 yrs.), MATHS LEVEL II (6-7 yrs.), SUPERSPY, BASIC TUTOR — BEGINNERS' LEVEL, ADVANCED LEVEL.	£3.50 each or two for £6.00
Dragon Educational Series : NUMBER CHASER	£2.00

PERIPHERALS

2nd hand floppy discs	50p each
SAM chips (74LS783)	£15.00 each

HOW TO ORDER
Write to us (or phone with an Access order) quoting the title, publisher and medium (where relevant) of the programs you would like. Please include 50p for postage/packaging on single orders, and 75p for two or more items ordered. Postage to Euope is £3.00 and £6.00 to rest of world.
We try to despatch within 24 hours, but allow up to 21 days for delivery if necessary.
Cheques/postal orders made payable to **JOHN PENN DISCOUNT SOFTWARE**.



JOHH PENN DISCOUNT SOFTWARE, DEAN FARM COTTAGE, KINGSLEY, BORDON, HANTS. GU35 9NG.
Tel. Bordon (04203) 5970
For all your Dragon hardware requirements contact
PEAKSOFT,
48 QUEEN ST., BALDERTON, NEWARK, NOTTS. NG24 3NS. Telephone. 0636 705230

DR55

TURBOCHARGE YOUR DRAGON:
With our great value hardware and software:

BASIC 42

Extended BASIC for the Dragon 64

For Dragondos (please state version) £14.95

Run your Dragon in 64K mode, while retaining BASIC and DOS. Print on hi-res screen, using standard PRINT commands, and a screen layout of 24 rows of 42 columns. Other features include:

Alternative, redefinable character sets, control key for special characters, repeating keys, and commands in lower case, windows, CATCH command for automatic return to window, inverted video (green on black/black on green), true underlining and extra PRINT commands and functions.

LIBRARY lists commands and functions. Automatic startup of BASIC program. TEXT command for software compatibility. Still 23335 bytes free to BASIC. Patches for Dragondos 1.0. Can load in extra UTILITIES from disk:

HELP UTILITY £5.00
Extensions to BASIC 42 include change cursor character, scroll disable, pause listing, BREAK disable, improved TRON (allows single stepping). Detailed help and error messages.

SPOOL UTILITY £5.00
Use computer while printing. 3.5K print buffer TYPIST program turns Dragon into typewriter.

NEW FOR '87:

ICONS UTILITY £5.00
Put icons in your program! Controlled by cursor or "mouse". Commands to define, clear, load and save icon positions and windows.

STRUCTUR UTILITY £5.00
Another first! Structured BASIC on the Dragon! Allows named procedures, improved loop controls by WHILE ... WEND, and REPEAT ... UNTIL etc.

DOS UTILITY £5.00
Make friends with your DOS! Enter all the main DOS commands, plus LIST, EDIT etc, and select files by cursor or "mouse".

WHAT THEY SAY ABOUT BASIC 42:

"probably the best step so far" ..... Dragon User

"an invaluable utility" ..... Dragon Update

HARDWARE

Memory Upgrades	**FROM**	**£35.00**
Superdos Cartridge		**£75.00**
Cumana Disk drives	**FROM**	**£140.00**
Superdos controller (chip only)		**£10.00**
Peaksoft joystick (ideal "mouse")		**£5.50**

DISK SOFTWARE FOR DRAGON 32/64/128 WITH DRAGONDOS/CUMANA DOS 2.0

• • NEW • •

Pixie (Mindsoft) £14.95
Icon-driven drawing program. Requires joystick.

• • •

DSKDREAM (Grosvenor) £19.95
The standard Dragon Editor/Assembler

D.R.S (Grosvenor) £9.95
Machine code database program

SOURCEMAKER (Pamcomms) £8.50
Disassembler for use with DSKDREAM

DISK-KIT (Pamcomms) £9.95
Sort out your disk problems

MACGOWAN SOFTWARE

PRINTER CONTROL* FROM £19.95
A text AND graphics processor

DUMPER* FROM £5.45
Relocatable screen dump program

COLOR PRINT* FROM £6.50
PMode 3 screen dump program

• • NEW • •

TROJAN LIGHTPEN S/W* CASS £7.00 / DISK £8.00
Upgrade includes screen dump
Complete with lightpen CASS £14.00 / DISK £16.00

MONITOR/ASSEMBLER* CASS £12.00 / DISK £15.00
Printer orientated

*Prices vary according to printer: please specify.

MONEYBOX (Harris) £14.99
Home and small business accounts

MAILBOX (Harris) £16.99
Selective mailing list program

SHAREBOX (Harris) £16.99
Manage your stocks and shares'

SALESBOX (Harris) £19.99
Balance B/F Sales Ledger'

BILLSBOX (Harris) £19.99
Balance B/F Purchase Ledger

CASHBOX (Harris) £19.99
Double-entry Nominal Ledger

STOCKBOX (Harris) £19.99
Full-featured Stock Control

ORDERBOX (Harris) £16.99
Invoicing linked to Sales or Stock

Cheques/P.O.'s/Further details/dealer enquiries to:

HARRIS MICRO SOFTWARE

49 Alexandra Road, Hounslow, Middlesex, TW3 4HP Tel: (01) 570 8335

DR57